Spice Grinder

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 app Namespace Reference

Classes

- class grindThread

Functions

- def toggle_motor (toggle, motor)
- def grindSpice (motor, amount)
- def hello (kwargs)
- def toggle (kwargs)
- def grindSpices (kwargs)
- def application (request)
- def cleanAndExit ()

Variables

- int Motor1 = 2

  THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.
- list Scale1 = [15,17]
- int Motor2 = 3
- list Scale2 = [18, 27]
- int Motor3 = 4
- list Scale3 = [22, 23]
- int Motor4 = 5
- list Scale4 = [24, 25]
- int Motor5 = 6
- list Scale5 = [19, 16]
- int Motor6 = 7
- list Scale6 = [26, 20]
- int Sleep = 12
- int Step = 13
- int Dir = 14
- int CW = 1
- int CCW = 0
- tuple Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir,Scale1[1],Scale2[1],Scale3[1],Scale4[1],Scale5[1],S
- tuple Ins = (Scale1[0],Scale2[0],Scale3[0],Scale4[0],Scale5[0],Scale6[0])
- dictionary motorDict

5.1.1 Function Documentation

5.1.1.1 def app.application (*request*)

5.1.1.2 def app.cleanAndExit ( )

5.1.1.3 def app.grindSpice (*motor, amount*)

5.1.1.4 def app.grindSpices (*kwargs*)

5.1.1.5 def app.hello (*kwargs*)

5.1.1.6 def app.toggle (*kwargs*)

5.1.1.7 def app.toggle_motor (*toggle, motor*)

5.1.2 Variable Documentation

5.1.2.1 int app.CCW = 0

5.1.2.2 int app.CW = 1

5.1.2.3 int app.Dir = 14

5.1.2.4 tuple app.Ins = (Scale1[0],Scale2[0],Scale3[0],Scale4[0],Scale5[0],Scale6[0])

5.1.2.5 int app.Motor1 = 2

THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.

5.1.2.6 int app.Motor2 = 3

5.1.2.7 int app.Motor3 = 4

5.1.2.8 int app.Motor4 = 5

5.1.2.9 int app.Motor5 = 6

5.1.2.10 int app.Motor6 = 7

5.1.2.11 dictionary app.motorDict

Initial value:

```
1 = {
2         '0': {
3             'motor': 2,
4             'scale': [15, 17],
5             'reference': 6659
6             },
7         '1': {
8             'motor': 3,
9             'scale': [18, 27]
10             },
11         '2': {
12             'motor': 4,
13             'scale': [22, 23],
14             'reference': 6921
15             },
16         '3': {
17             'motor': 5,
18             'scale': [24, 25]
19             },
20         '4': {
21             'motor': 6,
22             'scale': [19, 16]
23             },
24         '5': {
25             'motor': 7,
26             'scale': [26, 20]
27             },
28         }
```

5.1.2.12 tuple app.Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir,Scale1[1],Scale2[1],Scale3[1],Scale4[1],Scal

5.1.2.13 list app.Scale1 = [15,17]

5.1.2.14 list app.Scale2 = [18, 27]

5.1.2.15 list app.Scale3 = [22, 23]

5.1.2.16 list app.Scale4 = [24, 25]

5.1.2.17 list app.Scale5 = [19, 16]

5.1.2.18 list app.Scale6 = [26, 20]

5.1.2.19 int app.Sleep = 12

5.1.2.20 int app.Step = 13

5.2 buttonPressed Namespace Reference

Variables

- button = Button(18)

5.2.1 Variable Documentation

5.2.1.1 buttonPressed.button = Button(18)

5.3 conversion_func Namespace Reference

Functions

- def OunceToGram (ounce)
- def GramToOunce (gram)
- def KgToGram (kilogram)
- def GramToKg (gram)
- def PoundToGram (Pound)
- def PoundGramTo (gram)
- def TspToCup (tsp)
- def CupToTsp (Cup)
- def GramToTsp (cursor, gram, name)
- def TspToGram (cursor, tsp, name)

5.3.1 Function Documentation

5.3.1.1 def conversion_func.CupToTsp (*Cup*)

5.3.1.2 def conversion_func.GramToKg (*gram*)

5.3.1.3 def conversion_func.GramToOunce (*gram*)

5.3.1.4 def conversion_func.GramToTsp (*cursor, gram, name*)

5.3.1.5 def conversion_func.KgToGram (*kilogram*)

5.3.1.6 def conversion_func.OunceToGram (*ounce*)

5.3.1.7 def conversion_func.PoundGramTo (*gram*)

5.3.1.8 def conversion_func.PoundToGram (*Pound*)

5.3.1.9 def conversion_func.TspToCup (*tsp*)

5.3.1.10 def conversion_func.TspToGram (*cursor, tsp, name*)

5.4 database_func Namespace Reference

Functions

- def printSortedTable (cursor, tableName, sortBy)
- def printTable (cursor, tableName)
- def printTuple (cursor, tableName, attribute, field)
- def retrieveTuple (cursor, tableName, attribute, field)
- def insertNewRecipe (cursor, name, ingredient1, ingredient2, ingredient3, ingredient4, ingredient5, ingredient6, ingredient7, amount1, amount2, amount3, amount4, amount5, amount6, amount7)
- def insertNewSpice (cursor, name, gpt, available)
- def deleteTuple (cursor, tableName, attribute, field)

5.4.1 Function Documentation

5.4.1.1 def database_func.deleteTuple (*cursor, tableName, attribute, field*)

5.4.1.2 def database_func.insertNewRecipe (*cursor, name, ingredient1, ingredient2, ingredient3, ingredient4, ingredient5, ingredient6, ingredient7, amount1, amount2, amount3, amount4, amount5, amount6, amount7*)

5.4.1.3 def database_func.insertNewSpice (*cursor, name, gpt, available*)

5.4.1.4 def database_func.printSortedTable (*cursor, tableName, sortBy*)

5.4.1.5 def database_func.printTable (*cursor, tableName*)

5.4.1.6 def database_func.printTuple (*cursor, tableName, attribute, field*)

5.4.1.7 def database_func.retrieveTuple (*cursor, tableName, attribute, field*)

5.5 database_header Namespace Reference

Variables

- string deleteRecipeTable = 'DROP TABLE IF EXISTS ‘Recipe‘;'
- string deleteSpiceTable = 'DROP TABLE IF EXISTS ‘Spice‘;'
- string createSpiceTable
- string createRecipeTable

5.5.1 Variable Documentation

5.5.1.1 string database_header.createRecipeTable

5.5.1.2 string database_header.createSpiceTable

Initial value:

```
1 = '''CREATE TABLE Spice
2                         (
3                             id INTEGER PRIMARY KEY AUTOINCREMENT,
4                             name text NOT NULL DEFAULT '',
5                             grams_per_tsp real,
6                             available BIT DEFAULT 0
7 
8                         )
9                     '''
```

5.5.1.3 string database_header.deleteRecipeTable = 'DROP TABLE IF EXISTS ‘Recipe‘;'

5.5.1.4 string database_header.deleteSpiceTable = 'DROP TABLE IF EXISTS ‘Spice‘;'

5.6 database_main Namespace Reference

Variables

- conn = sqlite3.connect('BurrGrinder.db')
- c = conn.cursor()
- list spices

5.6.1 Detailed Description

```
see
https://docs.python.org/2/library/sqlite3.html
for some documentations on sqlite
```

5.6.2 Variable Documentation

5.6.2.1 database_main.c = conn.cursor()

5.6.2.2 database_main.conn = sqlite3.connect('BurrGrinder.db')

5.6.2.3 list database_main.spices

Initial value:

```
1 = [
2             (’Pepper’, 6.4, 1),
3             (’Salt’, 5.69, 1),
4             (’Paprika’, 1.6, 0),
5             (’Parsley’, 2.1, 0),
6             (’Cinnamon’, 2.3, 0),
7             (’Cloves’, 6.6, 0)
8          ]
```

5.7 example Namespace Reference

Functions

- def cleanAndExit ()
- def print_time (threadName, delay)

Variables

- hx = HX711(22, 23)
- val = hx.get_weight(5)

5.7.1 Function Documentation

5.7.1.1 def example.cleanAndExit ( )

5.7.1.2 def example.print_time (*threadName, delay*)

5.7.2 Variable Documentation

5.7.2.1 example.hx = HX711(22, 23)

5.7.2.2 example.val = hx.get_weight(5)

5.8 hx711 Namespace Reference

Classes

- class HX711

5.9 motors Namespace Reference

Functions

- def motor (toggle, motor)
- def steppermotor (rotation, direction)

Variables

- int Motor1 = 2

  THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.
- int Motor2 = 3
- int Motor3 = 4
- int Motor4 = 5
- int Motor5 = 6
- int Motor6 = 7
- int Sleep = 12
- int Step = 13
- int Dir = 14
- int CW = 1
- int CCW = 0
- tuple Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir)

5.9.1 Function Documentation

5.9.1.1 def motors.motor (*toggle, motor*)

5.9.1.2 def motors.steppermotor (*rotation, direction*)

5.9.2 Variable Documentation

5.9.2.1 int motors.CCW = 0

5.9.2.2 int motors.CW = 1

5.9.2.3 int motors.Dir = 14

5.9.2.4 int motors.Motor1 = 2

THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.

5.9.2.5 int motors.Motor2 = 3

5.9.2.6 int motors.Motor3 = 4

5.9.2.7 int motors.Motor4 = 5

5.9.2.8 int motors.Motor5 = 6

5.9.2.9 int motors.Motor6 = 7

5.9.2.10 tuple motors.Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir)

5.9.2.11 int motors.Sleep = 12

5.9.2.12 int motors.Step = 13

5.10 server Namespace Reference

Classes

- class grindThread

Functions

- def grind_spices ()
- def toggle_motor (toggle, motor)
- def grindSpice (motor, amount)

Variables

- app = Flask(__name__)
- methods
- int Motor1 = 2

  THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.
- list Scale1 = [15,17]
- int Motor2 = 3
- list Scale2 = [18, 27]
- int Motor3 = 4
- list Scale3 = [22, 23]
- int Motor4 = 5
- list Scale4 = [24, 25]
- int Motor5 = 6
- list Scale5 = [19, 16]
- int Motor6 = 7
- list Scale6 = [26, 20]
- int Sleep = 12
- int Step = 13
- int Dir = 14
- int CW = 1
- int CCW = 0
- tuple Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir,Scale1[1],Scale2[1],Scale3[1],Scale4[1],Scale5[1],S
- tuple Ins = (Scale1[0],Scale2[0],Scale3[0],Scale4[0],Scale5[0],Scale6[0])
- dictionary motorDict
- host
- debug

5.10.1 Function Documentation

5.10.1.1 def server.grind_spices ( )

5.10.1.2 def server.grindSpice (*motor, amount*)

5.10.1.3 def server.toggle_motor (*toggle, motor*)

5.10.2 Variable Documentation

5.10.2.1 server.app = Flask(__name__)

5.10.2.2 int server.CCW = 0

5.10.2.3 int server.CW = 1

5.10.2.4 server.debug

5.10.2.5 int server.Dir = 14

5.10.2.6 server.host

5.10.2.7 tuple server.Ins = (Scale1[0],Scale2[0],Scale3[0],Scale4[0],Scale5[0],Scale6[0])

5.10.2.8 server.methods

5.10.2.9 int server.Motor1 = 2

THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.

5.10.2.10 int server.Motor2 = 3

5.10.2.11 int server.Motor3 = 4

5.10.2.12 int server.Motor4 = 5

5.10.2.13 int server.Motor5 = 6

5.10.2.14 int server.Motor6 = 7

5.10.2.15 dictionary server.motorDict

Initial value:

```
1 = {
2          '0': {
3              'motor': Motor1,
4              'scale': Scale1,
5              'reference': 6659
6              },
7          '1': {
8              'motor': Motor2,
9              'scale': Scale2
10              },
11          '2': {
12              'motor': Motor3,
13              'scale': Scale3,
14              'reference': 6921
15              },
16          '3': {
17              'motor': Motor4,
18              'scale': Scale4
19              },
20          '4': {
21              'motor': Motor5,
22              'scale': Scale5
23              },
24          '5': {
25              'motor': Motor6,
26              'scale': Scale6
27              },
28          }
```

5.10.2.16 tuple server.Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir,Scale1[1],Scale2[1],Scale3[1],Scale4[1],S

5.10.2.17 list server.Scale1 = [15,17]

5.10.2.18 list server.Scale2 = [18, 27]

5.10.2.19 list server.Scale3 = [22, 23]

5.10.2.20 list server.Scale4 = [24, 25]

5.10.2.21 list server.Scale5 = [19, 16]

5.10.2.22 list server.Scale6 = [26, 20]

5.10.2.23 int server.Sleep = 12

5.10.2.24 int server.Step = 13

5.11 sqlite_ex Namespace Reference

Variables

- conn = sqlite3.connect('example.db')
- c = conn.cursor()
- tuple t = ('RHAT',)

5.11.1 Detailed Description

```
example code from
    https://docs.python.org/2/library/sqlite3.html
```

5.11.2 Variable Documentation

5.11.2.1 sqlite_ex.c = conn.cursor()

5.11.2.2 sqlite_ex.conn = sqlite3.connect('example.db')

5.11.2.3 tuple sqlite_ex.t = ('RHAT',)

Chapter 6

Class Documentation

6.1 app.grindThread Class Reference

Inheritance diagram for app.grindThread:

6.2 server.grindThread Class Reference

Inheritance diagram for server.grindThread:

Collaboration diagram for server.grindThread:

Public Member Functions

- def __init__ (self, motor, amount)
- def run (self)

Public Attributes

- motor
- amount

6.2.1 Constructor & Destructor Documentation

6.2.1.1 def server.grindThread.__init__ (*self, motor, amount*)

6.2.2 Member Function Documentation

6.2.2.1 def server.grindThread.run (*self*)

6.2.3 Member Data Documentation

6.2.3.1 server.grindThread.amount

6.2.3.2 server.grindThread.motor

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

- /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Motor-Code/src/server.py

6.3 hx711.HX711 Class Reference

Public Member Functions

- def __init__ (self, dout, pd_sck, gain=128)
- def is_ready (self)
- def set_gain (self, gain)
- def get_gain (self)
- def read (self)
- def get_binary_string (self)
- def get_np_arr8_string (self)
- def read_np_arr8 (self)
- def read_long (self)
- def read_average (self, times=3)
- def read_median (self, times=3)
- def get_value (self, times=3)
- def get_value_A (self, times=3)
- def get_value_B (self, times=3)
- def get_weight (self, times=3)
- def get_weight_A (self, times=3)
- def get_weight_B (self, times=3)
- def tare (self, times=15)
- def tare_A (self, times=15)
- def tare_B (self, times=15)
- def set_reading_format (self, byte_format="LSB", bit_format="MSB")
- def set_offset (self, offset)
- def set_offset_A (self, offset)
- def set_offset_B (self, offset)
- def set_reference_unit (self, reference_unit)
- def set_reference_unit_A (self, reference_unit)
- def set_reference_unit_B (self, reference_unit)
- def power_down (self)
- def power_up (self)
- def reset (self)
- def __init__ (self, dout, pd_sck, gain=128)
- def is_ready (self)
- def set_gain (self, gain)
- def get_gain (self)
- def read (self)
- def get_binary_string (self)
- def get_np_arr8_string (self)
- def read_np_arr8 (self)
- def read_long (self)
- def read_average (self, times=3)
- def read_median (self, times=3)
- def get_value (self, times=3)
- def get_value_A (self, times=3)
- def get_value_B (self, times=3)
- def get_weight (self, times=3)
- def get_weight_A (self, times=3)
- def get_weight_B (self, times=3)
- def tare (self, times=15)
- def tare_A (self, times=15)
- def tare_B (self, times=15)

- def set_reading_format (self, byte_format="LSB", bit_format="MSB")
- def set_offset (self, offset)
- def set_offset_A (self, offset)
- def set_offset_B (self, offset)
- def set_reference_unit (self, reference_unit)
- def set_reference_unit_A (self, reference_unit)
- def set_reference_unit_B (self, reference_unit)
- def power_down (self)
- def power_up (self)
- def reset (self)

Public Attributes

- PD_SCK
- DOUT
- GAIN
- REFERENCE_UNIT
- REFERENCE_UNIT_B
- OFFSET
- OFFSET_B
- lastVal
- isNegative
- MSBindex24Bit
- MSBindex32Bit
- LSByte
- MSByte
- MSBit
- LSBit
- byte_format
- bit_format
- byte_range_values
- bit_range_values

6.3.1 Constructor & Destructor Documentation

6.3.1.1 def hx711.HX711.__init__ (*self, dout, pd_sck, gain* = 128)

6.3.1.2 def hx711.HX711.__init__ (*self, dout, pd_sck, gain* = 128)

6.3.2 Member Function Documentation

6.3.2.1 def hx711.HX711.get_binary_string (*self*)

6.3.2.2 def hx711.HX711.get_binary_string (*self*)

6.3.2.3 def hx711.HX711.get_gain (*self*)

6.3.2.4 def hx711.HX711.get_gain (*self*)

6.3.2.5 def hx711.HX711.get_np_arr8_string (*self*)

6.3.2.6 def hx711.HX711.get_np_arr8_string (*self*)

6.3.2.7 def hx711.HX711.get_value (*self, times* = 3)

6.3.2.8 def hx711.HX711.get_value (*self, times* = 3)

6.3.2.9 def hx711.HX711.get_value_A (*self, times* = 3)

6.3.2.10 def hx711.HX711.get_value_A (*self, times* = 3)

6.3.2.11 def hx711.HX711.get_value_B (*self, times* = 3)

6.3.2.12 def hx711.HX711.get_value_B (*self, times* = 3)

6.3.2.13 def hx711.HX711.get_weight (*self, times* = 3)

6.3.2.14 def hx711.HX711.get_weight (*self, times* = 3)

6.3.2.15 def hx711.HX711.get_weight_A (*self, times* = 3)

6.3.2.16 def hx711.HX711.get_weight_A (*self, times* = 3)

6.3.2.17 def hx711.HX711.get_weight_B (*self, times* = 3)

6.3.2.18 def hx711.HX711.get_weight_B (*self, times* = 3)

6.3.2.19 def hx711.HX711.is_ready (*self*)

6.3.2.20 def hx711.HX711.is_ready (*self*)

6.3.2.21 def hx711.HX711.power_down (*self*)

6.3.2.22 def hx711.HX711.power_down (*self*)

6.3.2.23 def hx711.HX711.power_up (*self*)

6.3.2.24 def hx711.HX711.power_up (*self*)

6.3.2.25 def hx711.HX711.read (*self*)

6.3.2.26 def hx711.HX711.read (*self*)

6.3.2.27 def hx711.HX711.read_average (*self, times* = 3)

6.3.2.28 def hx711.HX711.read_average (*self, times* = 3)

6.3.2.29 def hx711.HX711.read_long (*self*)

6.3.2.30 def hx711.HX711.read_long (*self*)

6.3.2.31 def hx711.HX711.read_median (*self, times* = 3)

6.3.2.32 def hx711.HX711.read_median (*self, times* = 3)

6.3.2.33 def hx711.HX711.read_np_arr8 (*self*)

6.3.2.34 def hx711.HX711.read_np_arr8 (*self*)

6.3.2.35 def hx711.HX711.reset (*self*)

6.3.2.36 def hx711.HX711.reset (*self*)

6.3.2.37 def hx711.HX711.set_gain (*self, gain*)

6.3.2.38 def hx711.HX711.set_gain (*self, gain*)

6.3.2.39 def hx711.HX711.set_offset (*self, offset*)

6.3.2.40 def hx711.HX711.set_offset (*self, offset*)

6.3.2.41 def hx711.HX711.set_offset_A (*self, offset*)

6.3.2.42 def hx711.HX711.set_offset_A (*self, offset*)

6.3.2.43 def hx711.HX711.set_offset_B (*self, offset*)

6.3.2.44 def hx711.HX711.set_offset_B (*self, offset*)

6.3.2.45 def hx711.HX711.set_reading_format (*self, byte_format* = "LSB", *bit_format* = "MSB")

6.3.2.46 def hx711.HX711.set_reading_format (*self, byte_format* = "LSB", *bit_format* = "MSB")

6.3.2.47 def hx711.HX711.set_reference_unit (*self, reference_unit*)

6.3.2.48 def hx711.HX711.set_reference_unit (*self, reference_unit*)

6.3.2.49 def hx711.HX711.set_reference_unit_A (*self, reference_unit*)

6.3.2.50 def hx711.HX711.set_reference_unit_A (*self, reference_unit*)

6.3.2.51 def hx711.HX711.set_reference_unit_B (*self, reference_unit*)

6.3.2.52 def hx711.HX711.set_reference_unit_B (*self, reference_unit*)

6.3.2.53 def hx711.HX711.tare (*self, times* = 15)

6.3.2.54 def hx711.HX711.tare (*self, times* = 15)

6.3.2.55 def hx711.HX711.tare_A (*self, times* = 15)

6.3.2.56 def hx711.HX711.tare_A (*self, times* = 15)

6.3.2.57 def hx711.HX711.tare_B (*self, times* = 15)

6.3.2.58 def hx711.HX711.tare_B (*self, times* = 15)

6.3.3 Member Data Documentation

6.3.3.1 hx711.HX711.bit_format

6.3.3.2 hx711.HX711.bit_range_values

6.3.3.3 hx711.HX711.byte_format

6.3.3.4 hx711.HX711.byte_range_values

6.3.3.5 hx711.HX711.DOUT

6.3.3.6 hx711.HX711.GAIN

6.3.3.7 hx711.HX711.isNegative

6.3.3.8 hx711.HX711.lastVal

6.3.3.9 hx711.HX711.LSBit

6.3.3.10 hx711.HX711.LSByte

6.3.3.11 hx711.HX711.MSBindex24Bit

6.3.3.12 hx711.HX711.MSBindex32Bit

6.3.3.13 hx711.HX711.MSBit

6.3.3.14 hx711.HX711.MSByte

6.3.3.15 hx711.HX711.OFFSET

6.3.3.16 hx711.HX711.OFFSET_B

6.3.3.17 hx711.HX711.PD_SCK

6.3.3.18 hx711.HX711.REFERENCE_UNIT

6.3.3.19 hx711.HX711.REFERENCE_UNIT_B

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

- /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Scale_Code/hx711.py

Chapter 7

File Documentation

7.1 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/db/conversion_func.py File Reference

Namespaces

- conversion_func

Functions

- def conversion_func.OunceToGram (ounce)
- def conversion_func.GramToOunce (gram)
- def conversion_func.KgToGram (kilogram)
- def conversion_func.GramToKg (gram)
- def conversion_func.PoundToGram (Pound)
- def conversion_func.PoundGramTo (gram)
- def conversion_func.TspToCup (tsp)
- def conversion_func.CupToTsp (Cup)
- def conversion_func.GramToTsp (cursor, gram, name)
- def conversion_func.TspToGram (cursor, tsp, name)

7.2 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/db/database_func.py File Reference

Namespaces

- database_func

Functions

- def database_func.printSortedTable (cursor, tableName, sortBy)
- def database_func.printTable (cursor, tableName)
- def database_func.printTuple (cursor, tableName, attribute, field)
- def database_func.retrieveTuple (cursor, tableName, attribute, field)
- def database_func.insertNewRecipe (cursor, name, ingredient1, ingredient2, ingredient3, ingredient4, ingredient5, ingredient6, ingredient7, amount1, amount2, amount3, amount4, amount5, amount6, amount7)
- def database_func.insertNewSpice (cursor, name, gpt, available)
- def database_func.deleteTuple (cursor, tableName, attribute, field)

7.3 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/db/database_header.py File Reference

Namespaces

- database_header

Variables

- string database_header.deleteRecipeTable = 'DROP TABLE IF EXISTS ‘Recipe‘;'
- string database_header.deleteSpiceTable = 'DROP TABLE IF EXISTS ‘Spice‘;'
- string database_header.createSpiceTable
- string database_header.createRecipeTable

7.4 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/db/database_main.py File Reference

Namespaces

- database_main

Variables

- database_main.conn = sqlite3.connect('BurrGrinder.db')
- database_main.c = conn.cursor()
- list database_main.spices

7.5 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/db/sqlite_ex.py File Reference

Namespaces

- sqlite_ex

Variables

- sqlite_ex.conn = sqlite3.connect('example.db')
- sqlite_ex.c = conn.cursor()
- tuple sqlite_ex.t = ('RHAT',)

7.6 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Motor-Code/src/app.py File Reference

Classes

- class app.grindThread

Namespaces

- app

Functions

- def app.toggle_motor (toggle, motor)
- def app.grindSpice (motor, amount)
- def app.hello (kwargs)
- def app.toggle (kwargs)
- def app.grindSpices (kwargs)
- def app.application (request)
- def app.cleanAndExit ()

Variables

- int app.Motor1 = 2

  THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.
- list app.Scale1 = [15,17]
- int app.Motor2 = 3
- list app.Scale2 = [18, 27]
- int app.Motor3 = 4
- list app.Scale3 = [22, 23]
- int app.Motor4 = 5
- list app.Scale4 = [24, 25]
- int app.Motor5 = 6
- list app.Scale5 = [19, 16]
- int app.Motor6 = 7
- list app.Scale6 = [26, 20]
- int app.Sleep = 12
- int app.Step = 13
- int app.Dir = 14
- int app.CW = 1
- int app.CCW = 0
- tuple app.Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir,Scale1[1],Scale2[1],Scale3[1],Scale4[1],Scale5[
- tuple app.Ins = (Scale1[0],Scale2[0],Scale3[0],Scale4[0],Scale5[0],Scale6[0])
- dictionary app.motorDict

7.7 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Motor-Code/src/buttonPressed.py File Reference

Namespaces

- buttonPressed

Variables

- buttonPressed.button = Button(18)

7.8 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Motor-Code/src/motors.py File Reference

Namespaces

- motors

Functions

- def motors.motor (toggle, motor)
- def motors.steppermotor (rotation, direction)

Variables

- int motors.Motor1 = 2

  THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.
- int motors.Motor2 = 3
- int motors.Motor3 = 4
- int motors.Motor4 = 5
- int motors.Motor5 = 6
- int motors.Motor6 = 7
- int motors.Sleep = 12
- int motors.Step = 13
- int motors.Dir = 14
- int motors.CW = 1
- int motors.CCW = 0
- tuple motors.Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir)

7.9 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Motor-Code/src/server.py File Reference

Classes

- class server.grindThread

Namespaces

- server

Functions

- def server.grind_spices ()
- def server.toggle_motor (toggle, motor)
- def server.grindSpice (motor, amount)

Variables

- server.app = Flask(__name__)
- server.methods
- int server.Motor1 = 2

  THESE LINE WILL NEED TO BE ADDED TO THE MAIN CODE TO SETUP GPIO PINS########.
- list server.Scale1 = [15,17]
- int server.Motor2 = 3
- list server.Scale2 = [18, 27]
- int server.Motor3 = 4
- list server.Scale3 = [22, 23]
- int server.Motor4 = 5
- list server.Scale4 = [24, 25]
- int server.Motor5 = 6
- list server.Scale5 = [19, 16]
- int server.Motor6 = 7
- list server.Scale6 = [26, 20]
- int server.Sleep = 12
- int server.Step = 13
- int server.Dir = 14
- int server.CW = 1
- int server.CCW = 0
- tuple server.Outs = (Motor1,Motor2,Motor3,Motor4,Motor5,Motor6,Sleep,Step,Dir,Scale1[1],Scale2[1],Scale3[1],Scale4[1],Scale
- tuple server.Ins = (Scale1[0],Scale2[0],Scale3[0],Scale4[0],Scale5[0],Scale6[0])
- dictionary server.motorDict
- server.host
- server.debug

7.10 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Scale_Code/example.py File Reference

Namespaces

- example

Functions

- def example.cleanAndExit ()

Variables

- example.hx = HX711(22, 23)
- example.val = hx.get_weight(5)

7.11 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Motor-Code/src/example.py File Reference

Namespaces

- example

Functions

- def example.print_time (threadName, delay)

7.12 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Scale_Code/hx711.py File Reference

Classes

- class hx711.HX711

Namespaces

- hx711

7.13 /home/uvnahn21/Desktop/UTA/Fall 2018/cse 4316-005/burr grinder code/Motor-Code/src/hx711.py File Reference

Classes

- class hx711.HX711

Namespaces

- hx711

Index