

Arduino - Python

Generated by Doxygen 1.8.11

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	Counter.py File Reference	3
2.1.1	Function Documentation	3
2.1.1.1	calibDist(pin)	3
2.1.1.2	countPeople(distanceToCheck1, distanceToCheck2)	3
2.1.1.3	dbConnect()	4
2.1.1.4	insert(cnx, cursor, inOrOut)	4
2.1.1.5	readIn(pin)	4
	Index	5

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

Counter.py	3
----------------------------	-------	---

Chapter 2

File Documentation

2.1 Counter.py File Reference

Functions

- def `Counter.insert` (cnx, cursor, inOrOut)
- def `Counter.dbConnect` ()
- def `Counter.calibDist` (pin)
- def `Counter.readIn` (pin)
- def `Counter.countPeople` (distanceToCheck1, distanceToCheck2)
- def `Counter.signalHandler` (sig, frame)

Variables

- `Counter.board` = PyMata("/dev/ttyACM0", verbose=True)
- int `Counter.pin1` = 12
- int `Counter.pin2` = 13
- int `Counter.distToCheck1` = calibDist(pin1)-10
- int `Counter.distToCheck2` = calibDist(pin2)-10

2.1.1 Function Documentation

2.1.1.1 def Counter.calibDist (*pin*)

This function reads data from the sensors for a specified amount of time and then takes the average of those values to find the distance to check against for comparison (sensor-to-wall distance).
:return: integer

2.1.1.2 def Counter.countPeople (*distanceToCheck1*, *distanceToCheck2*)

This function reads data in from HC-SR04 (ultrasonic) sensors and registers a pedestrian entering/exiting based on proximity in comparison to "distanceToCheck" and which sensor is triggered first. It inserts this data into the MySQL database, passing in whether the pedestrian was entering or exiting.
:param distanceToCheck:
:return:

2.1.1.3 `def Counter.dbConnect ()`

Connects to the database using my credentials and the mysql password. Will print out a string depending on the status of the connection i.e.: whether it failed or not and why.
:return:

2.1.1.4 `def Counter.insert (cnx, cursor, inOrOut)`

This function inserts an entry into the database. Will insert the location (should be the Student Recreation Center but can be changed), a boolean value as to whether the subject exited or entered, the day of the week in number format (using the mysql `dayofweek(now())` function) and a DateTime stamp (using mysql `now()` function)
:param *cnx*:
:param *cursor*:
:param *inOrOut*: True=In, False=Out
:return: Boolean

2.1.1.5 `def Counter.readIn (pin)`

This function reads the data from the ultrasonic sensor associated with the passed-in Arduino pin for 1/10th of a second and calculates the average distance for those measurements. This is to provide more accurate data to reconcile the slight inaccuracy of the sensors. Note: ~100k measurements are taken in 1/10th of a second.
:param *pin*:
:return:

Index

- calibDist
 - Counter.py, [3](#)
- countPeople
 - Counter.py, [3](#)
- Counter.py, [3](#)
 - calibDist, [3](#)
 - countPeople, [3](#)
 - dbConnect, [3](#)
 - insert, [4](#)
 - readIn, [4](#)
- dbConnect
 - Counter.py, [3](#)
- insert
 - Counter.py, [4](#)
- readIn
 - Counter.py, [4](#)