PackagingGUI_2

index
/home/pi/Desktop/HX711_Python3/PackagingGUI_2.py

Modules

RPi.GPIO	mysql	tkinter.scrolledtext	_tkinter
datetime	os	sys	uuid
tkinter.messagebox	re	time	

Classes

Packaging

class **Packaging**

```
This class write the order number to the RFID tag and update the database information about the order.

Attribute
    @packageWeight
    @CALIBRATION (Tkinter Window): For calibrating the weight scale.
    @THRESHOLD (Tkinter Window): For entering the desired thresholds.
    @ROOT (Tkinter Window): The main window to check if the order exist.
    @cnx (MySQL Connection): Connection to MySQL database on the server.
```

Methods defined here:

Calibration(self)

```
This function reading in the package weight and write the order number to RFID tag.
Finally, it will envoke "Threshold" function

@hx (HX711): the weight scale;
@reader (SimpleMFRC522): RFID tag reader and writer.
```

CheckOrderOnDatabase(self)

```
This function retrieve the entered order number from a textfield, then check if it exist on the database.
It will show an error messagebox if the order number does not exist. Contrarily, it will display the contents on a t
Finally, it will invoke the "Calibration" function.
```

Main(self)

```
This function initialise the main window with labels, button, and entry field.

@ROOT (window): TKinter window with '400x200' scale.
@btnCheckOrder (entry): The button to invoke "CheckOrderOnDatabase" function
@txtOrderResults (entry): A text field to display the content of the order
```

Threshold(self)

```
This function display the "THRESHOLD" window for an user to enter the desired temperature.

@txtEnterTemperature (entry): The threshold temperature entry field.
@txtEnterHumidity (entry): The threshold humidity entry field.
```

destroy(self)

```
This function start by writing the order number to the tag, then save the inserted threshold values to the database.

@reader (SimpleMFRC522): RFID tag reader and writer.
```

Data and other attributes defined here:

CALIBRATION = None

ROOT = None

THRESHOLD = None

cnx = None

packageWeight = -1

reader = None

txtEnterHumidity = None

txtEnterOrderNumber = None

txtEnterTemperature = None

txtOrderResults = None

Data

ACTIVE = 'active'
ALL = 'all'
ANCHOR = 'anchor'
ARC = 'arc'
BASELINE = 'baseline'
BEVEL = 'bevel'

BOTH = 'both'
BOTTOM = 'bottom'
BROWSE = 'browse'
BUTT = 'butt'
CASCADE = 'cascade'
CENTER = 'center'
CHAR = 'char'
CHECKBUTTON = 'checkbutton'
CHORD = 'chord'
COMMAND = 'command'
CURRENT = 'current'
DISABLED = 'disabled'
DOTBOX = 'dotbox'
E = 'e'
END = 'end'
EW = 'ew'
EXCEPTION = 8
EXTENDED = 'extended'
FALSE = 0
FIRST = 'first'
FLAT = 'flat'
GROOVE = 'groove'
HIDDEN = 'hidden'
HORIZONTAL = 'horizontal'
INSERT = 'insert'
INSIDE = 'inside'
LAST = 'last'
LEFT = 'left'
MITER = 'miter'
MOVETO = 'moveto'
MULTIPLE = 'multiple'
N = 'n'
NE = 'ne'
NO = 0
NONE = 'none'
NORMAL = 'normal'
NS = 'ns'
NSEW = 'nsew'
NUMERIC = 'numeric'
NW = 'nw'
OFF = 0
ON = 1
OUTSIDE = 'outside'
PAGES = 'pages'
PIESLICE = 'pieslice'
PROJECTING = 'projecting'
RADIOBUTTON = 'radiobutton'
RAISED = 'raised'
READABLE = 2
RIDGE = 'ridge'
RIGHT = 'right'
ROUND = 'round'
S = 's'
SCROLL = 'scroll'
SE = 'se'
SEL = 'sel'
SEL_FIRST = 'sel.first'
SEL_LAST = 'sel.last'
SEPARATOR = 'separator'
SINGLE = 'single'
SOLID = 'solid'
SUNKEN = 'sunken'
SW = 'sw'
StringTypes = (<type 'str'>, <type 'unicode'>)
TOP = 'top'
TRUE = 1
TclVersion = 8.6
TkVersion = 8.6
UNDERLINE = 'underline'
UNITS = 'units'
VERTICAL = 'vertical'
W = 'w'
WORD = 'word'
WRITABLE = 4
X = 'x'
Y = 'y'
YES = 1
absolute_import = _Feature((2, 5, 0, 'alpha', 1), (3, 0, 0, 'alpha', 0), 16384)
wantobjects = 1