

SRRP-module controller

Generated by Doxygen 1.9.7

1 Description	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 main.Task Class Reference	7
4.1.1 Constructor & Destructor Documentation	7
4.1.1.1 __init__()	7
4.1.2 Member Function Documentation	8
4.1.2.1 pressDuration()	8
4.1.2.2 run()	8
4.1.2.3 startDelay()	8
4.2 main.Unit Class Reference	9
4.2.1 Constructor & Destructor Documentation	9
4.2.1.1 __init__()	9
4.2.2 Member Function Documentation	9
4.2.2.1 activate()	9
4.2.2.2 getPin()	11
5 File Documentation	13
5.1 main.py File Reference	13
5.1.1 Detailed Description	13
5.1.2 Author (s)	14
5.1.3 Function Documentation	14
5.1.3.1 CheckPhotoCell()	14
5.1.3.2 getWidthFromJSON()	14
Index	15

Chapter 1

Description

Python program for parallell controll of SRRP-modules. When a bag passes the photocell, a new task is created with an array of SRRP-units. It's then launced in parallel. Copyrigh (c) 2023 HOLDT - Hull og Lekkasje Deteksjons Teknologi. All rights reserved.

Chapter 2

Class Index

2.1 Class List

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

main.Task	7
main.Unit	9

Chapter 3

File Index

3.1 File List

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

main.py	Controlles the whole sequence	13
-------------------------	---	----

Chapter 4

Class Documentation

4.1 main.Task Class Reference

Public Member Functions

- `__init__` (self, ammount, pins)
Initializes a task.
- `run` (self, bag_size_in_m)
Runs task.
- `startDelay` (self, bag_size_in_m, is_first_iteration)
Calculates delay before a press starts.
- `pressDuration` (self, bag_size_in_m)
Calculates the duration of the press.

Public Attributes

- `ammount`
- `units`
- `pin_array`

4.1.1 Constructor & Destructor Documentation

4.1.1.1 `__init__()`

```
main.Task.__init__ (
    self,
    ammount,
    pins )
```

Initializes a task.

A Task is initialized with an array of **Unit** (s).

Parameters

<i>ammount</i>	The ammount of SRRP-modules in use.
<i>pin</i>	Array of pins that controll the modules.

Returns

GPIO pin-number.

4.1.2 Member Function Documentation**4.1.2.1 pressDuration()**

```
main.Task.pressDuration (
    self,
    bag_size_in_m )
```

Calculates the duration of the press.

Returns

Duration for the press.

4.1.2.2 run()

```
main.Task.run (
    self,
    bag_size_in_m )
```

Runs task.

Iterates through the task sequentially.

4.1.2.3 startDelay()

```
main.Task.startDelay (
    self,
    bag_size_in_m,
    is_first_iteration )
```

Calculates delay before a press starts.

Parameters

<i>bag_size_in_m</i>	Uses the bagsize as variable to calculate delay
<i>is_first_iteration</i>	Longer distance to first press, first iteration returns longer delay.

Returns

Delay for press to start

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

- [main.py](#)

4.2 main.Unit Class Reference

Public Member Functions

- `__init__` (self, pin)
Initializes a unit(single SRRP-moduel).
- `activate` (self, start_delay, press_duration)
Activates the module for a given duration.
- `getPin` (self)
Return the GPIO controll pin for the module.

Public Attributes

- `pin`

4.2.1 Constructor & Destructor Documentation

4.2.1.1 `__init__()`

```
main.Unit.__init__ (
    self,
    pin )
```

Initializes a unit(single SRRP-moduel).

Parameters

<i>self</i>	The module to controll.
<i>pin</i>	The GPIO pin number to controll the unit.

4.2.2 Member Function Documentation

4.2.2.1 `activate()`

```
main.Unit.activate (
    self,
    start_delay,
    press_duration )
```

Activates the module for a given duration.

Parameters

<i>self</i>	The module to controll.
<i>delay</i>	The ammount of time before the bag is in the correct position

4.2.2.2 getPin()

```
main.Unit.getPin (
    self )
```

Return the GPIO controll pin for the module.

Returns

GPIO pin-number.

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

- [main.py](#)

Chapter 5

File Documentation

5.1 main.py File Reference

Controlles the whole sequence.

Classes

- class [main.Unit](#)
- class [main.Task](#)

Functions

- [main.CheckPhotoCell](#) (old_state, new_state)
Checks the photocell if a new bag has passed.
- [main.getWidthFromJSON](#) (path)
Returns width of bag that's being analyzed.

Variables

- bool **main.old_state** = True
- **main.sensor** = GPIO.setup(157, GPIO.IN)
- list **main.pin_arr** = [74, 71, 73]
- **main.bag_passed**
- **main.tmp** = [Task](#)(len(pin_arr), pin_arr)
- int **main.bagWidth** = [getWidthFromJSON](#)('/home/radxa/SRRP-GUI/build/baginfo.json') / 1000
- **main.p** = Process(target=tmp.run, args=(bagWidth,)).start()

5.1.1 Detailed Description

Controlles the whole sequence.

5.1.2 Author (s)

- Eskild Dybwad Svenningsen
- Kristoffer Solheim

5.1.3 Function Documentation

5.1.3.1 CheckPhotoCell()

```
main.CheckPhotoCell (
    old_state,
    new_state )
```

Checks the photocell if a new bag has passed.

A new bag is considered to be in the system when it has fully passed the photocell. We check this by comparing the `old_state` to the `new_state`.

```
if old_state and old_state != new_state
```

Parameters

<i>old_state</i>	The previous state of the photocell.
<i>new_state</i>	The current state of the photocell.

Returns

(True, `new_state`) If bag has passed.

(False, `new_state`) If bag has not passed.

5.1.3.2 getWidthFromJSON()

```
main.getWidthFromJSON (
    path )
```

Returns width of bag that's being analyzed.

Parameters

<i>path</i>	Filepath to the json file containing bag information.
-------------	---

Returns

width Return width of given bag.

Index

- `__init__`
 - `main.Task`, [7](#)
 - `main.Unit`, [9](#)
- `activate`
 - `main.Unit`, [9](#)
- `CheckPhotoCell`
 - `main.py`, [14](#)
- `Description`, [1](#)
- `getPin`
 - `main.Unit`, [11](#)
- `getWidthFromJSON`
 - `main.py`, [14](#)
- `main.py`, [13](#)
 - `CheckPhotoCell`, [14](#)
 - `getWidthFromJSON`, [14](#)
- `main.Task`, [7](#)
 - `__init__`, [7](#)
 - `pressDuration`, [8](#)
 - `run`, [8](#)
 - `startDelay`, [8](#)
- `main.Unit`, [9](#)
 - `__init__`, [9](#)
 - `activate`, [9](#)
 - `getPin`, [11](#)
- `pressDuration`
 - `main.Task`, [8](#)
- `run`
 - `main.Task`, [8](#)
- `startDelay`
 - `main.Task`, [8](#)