2022

Selecta Shield Library Documentation

USING THE SELECTASHIELDLIBRARY WITH PYTHON

TIMO WIEDMER

AGTATEC AG | BZU

1 Table of Contents

2 Basic Setup

2.1 Setting up the library

The SelectaShieldLibrary can be found in the following GitHub repository. In order to use this module in your python project you need to either copy the python file into the directory of your python project or add it to the PYTHONPATH environment variable.

2.2 Importing the module into your python project

Now that you've downloaded the module and set up your environment for using the SelectaShieldLibrary you need to import the module into your python project. This can be done as following:

```
import SelectaShieldLibrary
```

3 Using the SelectaShieldLibrary

The class controlling the Selecta Shield is called SelectaPi

3.1 Initializing a new SelectaPi object with default parameters

A new SelectaPi object is initialized as following:

```
SelectaShield = SelectaShieldLibrary.SelectaPi()
```

3.2 Initializing a new SelectaPi object with custom parameters

The initialization function takes 2 parameters:

3.2.1 HomeAllMotors Parameter

HomeAllMotors defines, whether all the motors should return to their home position if they aren't in their home position already. If all motors are in their home position on initialization, no motor will be turned.

If HomeAllMotors is set to True, all motors will return to their home position.

If HomeAllMotors is set to False, no motor will return to their home position, even if they aren't in their home position.

The default value of HomeAllMotors is True

3.2.2 DrinkNames Parameter

DrinkNames must be a tuple containing 6 strings. These 6 strings will be assigned to the 6 soda slots of the vending machine. These names can later be used to get slot numbers by their names.

The default name of the slots is an empty string.

3.2.3 Initialization Examples

The Syntax for function parameters as shown is the same everywhere.

If you set a function's parameters by its parameter name, the order of the parameters doesn't matter. However, if you set the parameters without using the parameter name,

3.2.3.1 Initialization with HomeAllMotors

This initialization can be done as following:

```
SelectaShield = SelectaShieldLibrary.SelectaPi(True)
```

Or

```
SelectaShield = SelectaShieldLibrary.SelectaPi(HomeAllMotors=True)
```

3.2.3.2 Initialization with DrinkNames

This initialization can be done as following:

```
SelectaShield = SelectaShieldLibrary.SelectaPi(DrinkNames=("Cola", "Cola",
"Sprite", "Fanta", "Valser", "Valser"))
```

3.2.3.3 Initialization with both

This initialization can be done as following:

```
SelectaShield = SelectaShieldLibrary.SelectaPi(True, ("Cola", "Cola",
"Sprite", "Fanta", "Valser", "Valser"))
```

Or

```
SelectaShield = SelectaShieldLibrary.SelectaPi(HomeAllMotors=True,
DrinkNames=("Cola", "Cola", "Sprite", "Fanta", "Valser", "Valser"))
```

4 SelectaPi Class

4.1 Variables

4.1.1 SlotNames

SlotNames contains the names of the slots, which were set on initialization

4.1.2 Slot Level Variables

FULL, EMPTY

This is used together with the checkSlotLevel and checkSlotLevels Methods. These variables can be used for comparison. When using these variables, you don't have to change anything in your code, in case these values change in the module. FULL symbols that the Soda can switch is pressed. EMPTY states that the magazine is empty or will be empty soon.

4.1.3 Switch Function Variables

ENCODER_BUTTON, LEVEL_BUTTON, SELECT_BUTTON

The switch function variables are used together with the readButton and readButtons methods. They select, which set of buttons to read.

4.1.4 ON/OFF Variables

ON, OFF

The ON and OFF variables can be used with every read and write(set) function.

4.2 List of all Methods

```
readData()
getSlotName(SlotNumber)
getSlotNumber(SlotName)
setSlotName(SlotNumber, SlotName)
checkSlotLevels()
checkSlotLevel(SlotNumber)
readButtons(SwitchFunction)
readButton(SwitchFunction, ButtonNum)
setLEDs(LEDValues)
setLED(LEDValue, LEDNum)
setMotors(MotorValues)
setMotor(MotorValue, MotorNum)
executeMotorCycles(Cycles, MotorNum)
```

4.3 Method Descriptions

4.3.1 readData()

Arguments: none

Reads the values of the databus and returns them as a list.

4.3.2 getSlotName()

Arguments: SlotNumber

Returns the name of a Slot specified by the *SlotNumber*.

4.3.3 getSlotNumber()

Arguments: SlotName

Returns the *SlotNumber* of the first Slot with the specified *SlotName*.

4.3.4 setSlotName()

Arguments: SlotNumber, SlotName

Sets the name of the Slot with the specified *SlotNumber* to the specified *SlotName*.

4.3.5 checkSlotLevels()

Arguments: none

Reads the values of the level switches. Returns a list containing the values of all level switches

4.3.6 checkSlotLevel()

Arguments: SlotNumber

Reads the values of the level switches. Returns the value of the level switch placed in the Slot with the specified *SlotNumber*.

4.3.7 readButtons()

Arguments: SwitchFunction

Reads the values of the set of buttons specified by the *SwitchFunction*. Returns a list containing the values of the set of buttons.

4.3.8 readButton()

Arguments: SwitchFunction, ButtonNum

Reads the values of the set of buttons specified by the *SwitchFunction*. Returns the value of the button specified by the *ButtonNum*.

4.3.9 setLEDs()

Arguments: LEDValues

Sets the LEDs to the specified *LEDValues*. *LEDValues* must be a list with at least 6 items. Any additional items will be ignored.

4.3.10 setLED()

Arguments: LEDValue, LEDNum

Sets the LED specified with *LEDNum* to the specified *LEDValue*.

4.3.11 setMotors()

Arguments: MotorValues

Sets the Motors to the specified *MotorValues*. *MotorValues* must be a list with at least 6 items. Any additional items will be ignored.

4.3.12 setMotor()

Arguments: MotorValue, MotorNum

Sets the Motor specified with *MotorNum* to the specified *MotorValue*.

4.3.13 executeMotorCycles()

Arguments: Cycles, MotorNum

Executes an amount of dispense cycles specified by *Cycles* with the motor specified with *MotorNum*.