

## SLMPeriodicTester

```
+ measurementProgress: pyqtSignal «class attribute»
+ calibrationProgress: pyqtSignal «class attribute»
+ realTimeValues: pyqtSignal «class attribute»
+ timerStarted: pyqtSignal «class attribute»
+ loggingMsg: pyqtSignal «class attribute»
# ldaModel: LinearDiscriminantAnalysis «class attribute»
# kpcaModel: KerneLPCA «class attribute»
# gnbClassifier: GaussianNB «class attribute»
# weightingFactors: pd.DataFrame «class attribute»
# acceptanceLimits: pd.DataFrame «class attribute»
# resource_manager: pyVisa.ResourceManager «instance attribute»
# AFG: pyVisa.GPIBInstrument «instance attribute»
# DMM: pyVisa.GPIBInstrument «instance attribute»
# calibration_consecutive: str «instance attribute»
# dut: SoundLevelMeter «instance attribute»
# customer_info: dict «instance attribute»
# ambient_condition_values: dict «instance attribute»
# result_box: tuple «instance attribute»
# calibration_stage: int «instance attribute»
# calibration_state: int «instance attribute»
# power_supply_results: np.array «instance attribute»
+ fweighting_results: dict «instance attribute»
+ current_frequency: int «instance attribute»
+ freetime_1kHz_results: dict «instance attribute»
+ ref_linearity_results: dict «instance attribute»
+ elect_stab_time: float «instance attribute»
```

```
+ train() → None «instance method»
+ run_main_sequence() → None «instance method»
+ check_state() → None «instance method»
+ set_dut(dut: SoundLevelMeter) → None «instance method»
+ set_customer_info(customer_info: dict) → None «instance method»
+ set_consecutive(consecutive: str) → None «instance method»
+ set_standards(DMM:dict, AA: dict) → None «instance method»
+ detect_result() → None «instance method»
+ test_power_supply() → float «instance method»
+ test_electrical_fweightings(): int=30) → None «instance method»
+ test_freetime_weightings_1kHz():
int=30) → None «instance method»
+ set_state(): int=30) → None «instance method»
+ set_stage(stage: int) → None «instance method»
+ estimate_expanded_uncertainty() → None «instance method»
+ report_ambient_conditions(temperature:float,
humidity:float, pressure: float) → None «instance method»
+ read_screen() → None «instance method»
+ volt2level(v:float, ref_lev: tuple) → float «static
method»
```

## SoundLevelMeter

```
# info: dict «instance attribute»
# cl: int «instance attribute»
# preamplifier: dict «instance attribute»
# mic: dict «instance attribute»
# reference_level: float «instance attribute»
# calibration_results: list «instance attribute»
```

```
+ set_power_supply_values(values:
np.ndarray) → None «instance method»
+ set_calibration_check_indications(init_adjustment:
float, indications: np.array) → None «instance method»
+ set_reference_voltages_values(voltages: np.ndarray)
→ None «instance method»
```

«instantiate»

«instantiate»

## GUIController

```
# TESTER: SLMPeriodicTester «instance attribute»
# gui: SonometersCalibrationUI «instance attribute»
# resource_manager: pyVisa.ResourceManager «instance attribute»
+ calibrationThread: PyQt5.QThread «instance attribute»
+ save_standards_state: bool «instance attribute»
+ save_DUT_info_sate: bool «instance attribute»
+ AFG_info: dict «instance attribute»
+ DecadeBox_info: dict «instance attribute»
+ DMM_info: dict «instance attribute»
+ self_test_passed: bool «instance attribute»
+ electric_ff_corrections: pd.DataFrame «instance attribute»
+ instruction: QMessageBox «instance attribute»
+ searchStandardWorker: PyQt5.QObject «instance attribute»
+ searchStandardThread: PyQt5.QThread «instance attribute»
+ selfTesterWorker: PyQt5.QObject «instance attribute»
+ selfTesterThread: PyQt5.QThread «instance attribute»
+ cameraWorker: PyQt5.QObject «instance attribute»
+ cameraThread: PyQt5.QThread «instance attribute»
+ streaming_running: bool «instance attribute»
+ x1, y1, x2, y2: int «instance attributes»
+ timer_started: bool «instance attribute»
+ frames: list «instance attribute»
+ frames_queue: Queue «instance attribute»
+ imgProcessingThread: ImageProcessingThread «instance
attribute»
```

```
# connect_signals() → None «instance method»
+ save_dut_info() → None «instance method»
+ print_logging_msg(msg: str) → None «instance method»
+ save_standards_info() → None «instance method»
+ read_elec_ff_corrections() → None «instance method»
+ save_dut_info() → None «instance method»
+ search_standards() → None «instance method»
+ fill_standards_info() → None «instance method»
+ self_test() → None «instance method»
+ show_self_test_results(results: tuple) → None «instance method»
+ streaming_control() → None «instance method»
+ stream_frame() → None «instance method»
+ capture_frames() → None «instance method»
+ start() → None «instance method»
+ pause() → None «instance method»
+ run_sequence() → None «instance method»
+ sequence_control() → None «instance method»
+ eval_answer(instruction:
PyQt5.QMessageBox) → None «instance method»
+ parallel_dialog_response(answer) → None «instance method»
+ show_power_supply_values() → None «instance method»
+ save_cal_ind_values() → None «instance method»
+ save_ref_volt() → None «instance method»
+ img2qixmap() → None «instance method»
```

«instantiate»

## QObject

## StandardsSearcher

```
+ finished: pyqtSignal «class attribute»
+ progress: pyqtSignal «class attribute»
# resources_manager: visa.ResourceManager
«instance attribute»
+ DMM_model: str «instance attribute»
+ AA_model: str «instance attribute»
+ resources: tuple «instance attribute»
+ data: dict «instance attribute»
```

```
+ search() → None «instance method»
```

«instantiate»

## SelfTester

```
+ finished: pyqtSignal «class attribute»
+ progress: pyqtSignal «class attribute»
+ loggingMsg: pyqtSignal «class attribute»
# DMM: visa.Resource «instance attribute»
# AA: visa.Resource «instance attribute»
```

```
+ self_test() → None «instance method»
```

«instantiate»