

List of commands (public functions) of the SI1145_WE library

Function	Parameters	what it does
<code>void Init()</code>	none	initiates the SI1145 with some default register values
<code>void resetSI1145()</code>	none	reset of the device
<code>void setI2CAddress(<i>address</i>)</code>	7 Bit I2C Address	changes the I2C address from default (0x60)
<code>void enableMeasurements(<i>type, mode</i>)</code>	type: PS, ALS, PSALS, ALSUV, PSALSUV mode: FORCE, AUTO, PAUSE	configures channel list, starts / pauses measurement(s)
<code>void startSingleMeasurement()</code>	none	starts a single measurement (forced mode) of the types which have been enabled (enableMeasurements)
<code>void enableInterrupt(<i>type</i>)</code>	PS_INT, ALS_INT, PSALS_INT	enables interrupts; interrupt occurs when measurement is completed
<code>void disableAllInterrupts()</code>	none	disables all interrupts
<code>void setMeasurementRate(<i>rate</i>)</code>	0....65535	SI1145 wakes up periodically - a rate of x means that every x wake up a measurement is done
<code>void setLEDCurrent(<i>level</i>)</code>	1....15	sets the current for the IR LED; see datasheet page 5 for translation of level into mA
<code>void selectPsDiode(<i>diode</i>)</code>	SMALL_DIODE, LARGE_DIODE	selection of the photodiode for proximity measurements
<code>void selectIrDiode(<i>diode</i>)</code>	SMALL_DIODE, LARGE_DIODE	selection of the photodiode for infrared light measurements
<code>void enableHighResolutionPs()</code> <code>void disableHighResolutionPs()</code>	none	The A/D converter has 17 bit. If enabled, the 16 LSBs are used. Default: disabled - 16 MSBs are used.
<code>void enableHighResolutionVis()</code> <code>void disableHighResolutionVis()</code>	none	The A/D converter has 17 bit. If enabled, the 16 LSBs are used. Default: disabled - 16 MSBs are used.
<code>void enableHighResolutionIr()</code> <code>void disableHighResolutionIr()</code>	none	The A/D converter has 17 bit. If enabled, the 16 LSBs are used. Default: disabled - 16 MSBs are used.
<code>void setPsAdcGain(<i>factor</i>)</code>	0 ... 5	increases the LED pulse width by 2 ^{factor} , default is 0
<code>void enableHighSignalPsRange()</code> <code>void disableHighSignalPsRange()</code>	none	PS gain is divided by 14.5; used to prevent ADC overflow e.g. for measurements in direct sunlight
<code>void enableHighSignalVisRange()</code> <code>void disableHighSignalVisRange()</code>	none	ALS VIS/IR gain is divided by 14.5; used to prevent ADC overflow e.g. for measurements in direct sunlight
<code>void setAlsVisAdcGain(<i>factor</i>)</code>	0 ... 7	increases the integration time (measurement time) for ALS Vis by 2 ^{factor}
<code>void setAlsIrAdcGain(<i>factor</i>)</code>	0 ... 7	increases the integration time (measurement time) for ALS IR by 2 ^{factor}
<code>uint16_t getAlsVisData()</code>	none	returns the 16 bit measurement data for ALS VIS
<code>uint16_t getAlsIrData()</code>	none	returns the 16 bit measurement data for ALS IR
<code>uint16_t getPpData()</code>	none	returns the 16 bit measurement data for PS
<code>float getUvIndex()</code>	none	returns the UV - Index
<code>uint8_t getFailureMode()</code>	none	returns the content of the response register
<code>void clearFailure()</code>	none	clears response register
<code>void clearAllInterrupts()</code>	none	clears all interrupts
<code>void clearAlsInterrupt()</code>	none	clears ALS (VIS/IR) interrupts
<code>void clearPsInterrupt()</code>	none	clears PS interrupts
<code>byte clearCmdInterrupt()</code>	none	clears Interrupts triggered by Commands
<code>uint8_t getInterruptStatus();</code>	none	returns the content of the interrupt status register

ALS = ambient light
IR = infrared
PS = proximity
VIS = visual