List of commands (public functions) of the INA219_WE library

Function	Parameters	what it does
bool Init( )	none	initiates the INA219 with some default register values; returns true, if the INA219 is connected.
void reset_INA219( )	none	reset of the device
void setCorrectionFactor(*factor*)	factor (float)	if INA226 current values differ from currents measured with calibrated equipment, you can apply a factor
void setADCMode(*mode*)	BIT_MODE_9 BIT_MODE_10 BIT_MODE_11 BIT_MODE_12 SAMPLE_MODE_2 SAMPLE_MODE_4 SAMPLE_MODE_8 SAMPLE_MODE_16 SAMPLE_MODE_32 SAMPLE_MODE_64 SAMPLE_MODE_128	sets the ADC mode for shunt and bus voltage conversion BIT_MODE_X: single measurement with x bit resolution SAMPLE_MODE_X: average of X measurements
void setMeasureMode(*mode*)	POWER_DOWN TRIGGERED ADC_OFF CONTINUOUS	sets continuous or triggered mode, but also power down or switches ADC off for POWER_DOWN please chose "powerDown" function since it saves the configuration
void setPGain(*gain*)	PG_40 PG_80 PG_160 PG_320	sets the PGain value; the number is the maximum shunt voltage in mV. Using PG_320 and a 0.1 Ohm shunt sets the current range to 3.2 amperes.
void setBusRange(*mode*)	BRNG_16 BRNG_32	bus voltage range 0-16 Volt / 0 - 32 Volt
void setShuntSizeInOhms(*size*)	shuntSizeInOhms (float)	Define the shunt size in case you don't use a shunt of 0.1 ohms, which is the standard on modules.
void setShuntVoltOffset_mV(*offset*)	offset in millivolts (float)	People have reported offsets, i.e. there is a shunt voltage although there is no current (load switched off). The function will correct the current and power.
float getShuntVoltage_mV( )	none	delivers shunt voltage in mV
float getBusVoltage( )	none	delivers bus voltage in mV
float getCurrent_mA( )	none	delivers current in mA
float getBusPower( )	none	delivers the power in mW
bool getOverflow( )	none	delivers "true" if an overflow occurs in one of the data registers
void startSingleMeasurement( )	none	starts single shot measurement and waits until data is available
void powerDown( )	none	switches the module off and saves the configuration before
void powerUp( )	none	switches the module on after Power Down and writes back the configuration (modes, gains, etc)