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*)	INA219_BIT_MODE_9 INA219_BIT_MODE_10 INA219_BIT_MODE_11 INA219_BIT_MODE_12 INA219_SAMPLE_MODE_2 INA219_SAMPLE_MODE_4 INA219_SAMPLE_MODE_8 INA219_SAMPLE_MODE_16 INA219_SAMPLE_MODE_32 INA219_SAMPLE_MODE_64 INA219_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*)	INA219_POWER_DOWN INA219_TRIGGERED INA219_ADC_OFF INA219_CONTINUOUS + bus voltage only an current only versions	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*)	INA219_PG_40 INA219_PG_80 INA219_PG_160 INA219_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*)	INA219_BRNG_16 INA219_BRNG_32	Bus voltage range 0-16 Volt / 0 - 32 Volt
void setShuntSizeInOhms(*size*)	shuntSizeInOhms (float)	Defines the shunt size in case you don't use a shunt of 0.1 ohms. You find 0.1 ohm shunts on most 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	Returns shunt voltage in mV
float getBusVoltage( )	none	Returns bus voltage in mV
float getCurrent_mA( )	none	Returns current in mA
float getBusPower( )	none	Returns the power in mW
bool getOverflow( )	none	Returns "true" if an overflow occurs in one of the data registers
void startSingleMeasurement(*timeout*)	optional: timeout	Starts single shot measurement and waits until data is available (blocking). If timeout is passed to the function and if it is exceeded the sketch will return from the function.
void startSingleMeasurementNoWait( )	none	Starts single shot measurement. It does not wait until data is available (non-blocking).
bool getConversionReady( )	none	Returns true if a conversion is ready.
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).