

List of commands (public functions) of the INA226_WE library

| Function | Parameters | what it does |
|--|---|--|
| <code>void Init()</code> | none | initiates the INA226 with some default register values |
| <code>void reset_INA226()</code> | none | reset of the device |
| <code>void getI2cErrorCode()</code> | none | returns the current error code from endTransmission(); 0 = success. |
| <code>void setCorrectionFactor(<i>factor</i>)</code> | factor (float) | if INA226 current values differ from currents measured with calibrated equipment, you can apply a factor |
| <code>void setAverage(<i>mode</i>)</code> | INA226_AVERAGE_X X = 1, 4, 16, 64, 128, 256, 512, 1024 | sets the number of samples that are averaged for one measurement |
| <code>void setConversionTime(<i>time</i>)</code> | INA226_CONV_TIME_X X = 140, 204, 332, 588, 1100, 2116, 4156, 8244 | sets time for conversion for shunt and bus voltage in microseconds |
| <code>void setMeasureMode(<i>mode</i>)</code> | INA226_CONTINUOUS, INA226_TRIGGERED, INA226_POWER_DOWN + current only or bus voltage only versions | sets the mode; for POWER_DOWN please use the powerDown function since it remembers the mode before power-down; see also the examples. |
| <code>void setResistorRange(<i>resistorValue</i>, <i>range</i>)</code> | resistorValue in ohms (float), range in Ampere (float) | Sets resistor value in case you don't use an INA226 module with 0.1 ohms shunt. The second parameter (range) is optional (see example). |
| <code>float getShuntVoltage_mV()</code> | none | delivers shunt voltage in mV |
| <code>float getBusVoltage()</code> | none | delivers bus voltage in V |
| <code>float getCurrent_mA()</code> | none | delivers current in mA |
| <code>float getBusPower_mW()</code> | none | delivers the power in mW |
| <code>void startSingleMeasurement()</code> | none | starts single shot measurement and waits until data is available |
| <code>void startSingleMeasurementnoWait()</code> | none | starts single shot measurement and does not wait (non-blocking) |
| <code>void powerDown()</code> | none | switches the module off and saves the configuration before |
| <code>void powerUp()</code> | none | switches the module on after Power Down and writes back the configuration (modes, gains, etc) |
| <code>void waitUntilConversionCompleted()</code> | none | waits until the current conversions and calculations are completed. |
| <code>void setAlertPinActiveHigh()</code> | none | by default the alert pin is active-low; this function changes this |
| <code>void enableAlertLatch()</code> | none | the alert flag is set and the alert pin is active, when the limit in the alert register is exceeded; by default it will be deleted with the next measurement in limit; with enableAlertLatch the flag will have to be cleared manually, which gives better control |
| <code>void setAlertType(<i>type</i>, <i>limit</i>)</code> | types: INA226_SHUNT_UNDER, INA226_SHUNT_OVER, INA226_BUS_UNDER, INA226_BUS_OVER, INA226_CURRENT_UNDER, INA226_CURRENT_OVER, INA226_POWER_OVER limit: float | sets the alert type and the limit: SHUNT_OVER/_UNDER: limit in mV BUS_OVER / _UNDER: limit in V CURRENT_OVER / _UNDER: limit in mA POWER_OVER: limit in mW |
| <code>void readAndClearFlags()</code> | none | reads the Mask/Enable register; this clears the overflow, conversion ready and limit alert flags; the status of the flags are saved in the following bool variables: - overflow - convAlert - limitAlert |