List of commands (public functions) of the MCP23017 library

Function	what it does
void Init();	initiates the MCP23017 with some register values and sets some private variables
void reset();	reset of the MCP23107
void setPinMode(*pin, port, state*);	sets INPUT/OUTPUT/INPUT_PULLUP for a single pin
void setPortMode(*val, port*);	sets INPUT/OUTPUT for a complete port
void setPortMode(*val, port, INPUT_PULLUP*);	with this variant input pins are pulled up; no effect on output pins
void setPin(*pin, port, state*);	LOW/HIGH for a single pin
void togglePin(*pin, port*);	switches LOW to HIGH or HIGH to LOW
void setPinX(*pin, port, val, state*);	combination of setPin and setPinMode
void setAllPins(*port, state*);	switches all Pins to HIGH or LOW (all same)
void setPort(*val, port*);	sets HIGH/LOW for all Pins
void setPort(*val, val*);	sets HIGH / LOW for pins of both ports (A, B);
void setPortX(*val, val, port*);	sets pinMode and HIGH/LOW for a complete port (combination of setPortMode and setPort)
void setInterruptPinPol(*state*);	sets the polarity of INTA and INTB (active-high or active-low)
void setIntOdr(*state*);	sets INTA and INTB as open drain
void setInterruptOnChangePin(*pin, port*);	sets interrupt-on-change for a single pin
void setInterruptOnDefValDevPin(*pin, port, state*);	sets interrupt-on-defval-deviation a single pin
void setInterruptOnChangePort(*val, port*);	sets interrupt-on-change for a port
void setInterruptOnDefValDevPort(*val, port, val*);	sets interrupt-on-defval-deviation a single pin
void deleteAllInterruptsOnPort(*port*);	interrupt pins turn into "normal" pins
void setPinPullUp(*pin, port, state*);	sets internal pull-up for a single pin (only input pins are affected)
void setPortPullUp(*val, port*);	sets internal pull-up for a port (only input pins are affected)
void setIntMirror(*state*);	0/OFF: INTA and INTB working separately; 1/ON: INTA and INTB are mirrored
byte getIntFlag(*port*);	provides the content of the INTFLAG register
bool getPin(*pin, port*);	provides the logic level of a single pin
byte getPort(*port*);	provides the logic level of a port
byte getIntCap(*port*);	provides the content of the interrupt capture register

pin : pin number as byte
state : OFF or ON, or: LOW / HIGH, 0/1, INPUT, OUTPUT and INPUT_PULLUP
port : A or B (see PORT enum definition)
val : value of a Port or Register (byte)