

List of commands (public functions) of the MCP23017 library

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