



- ◇ `_mqttCurrentState : ePicoWatchState`
- ◇ `_mqttOldState : ePicoWatchState`
- ◇ `_currentState : ePicoWatchState`
- ◇ `evInitial_ : PicoEvent`
- ◇ `evDefault_ : PicoEvent`
- ◇ `evRelease_ : PicoEvent`
- ◇ `evLongPress_ : PicoEvent`
- ◇ `evConnectionTimeout_ : PicoEvent`
- ◇ `conAttemptcounter_ : uint8_t`
- ◇ `isLteConnectProcessActive_ : bool`
- ◇ `isMqttConnectProcessActive_ : bool`
- ◇ `isLteActive_ : bool`
- ◇ `connectionKeepAliveCounter_ : uint8_t`
- ◇ `evConnected_ : PicoEvent`
- ◇ `evError_ : PicoEvent`
- ◇ `_currentEvent : EvElement`
- ◇ `_currentButtonAction : uint8_t`
- ▣ `picoWorks[application::PicoWatch::WRK_NR_WORKS] : struct k_work`
- ▣ `picoTimers[application::PicoWatch::T_NR_TIMERS] : struct k_work`
- ▣ `_ledA; : gpio::LedController*`
- ▣ `_ledB : gpio::LedController*`
- ▣ `_ledC : gpio::LedController*`
- ▣ `nbrButtons = BUTTONS_NUMBER : const uint8_t`
- ▣ `_buttons[_nbrButtons] : gpio::Button*`
- ▣ `mqttData : string`
- ▣ `_gnssLocation : Location_t`
- ▣ `_gnssDateTime : GnssController::gDate_t`
- ▣ `_updateGnssPeriode : uint32_t`
- ▣ `_updateMqttPeriode : uint32_t`
- ▣ `_isMotorOn : bool`
- ▣ `batLvlMvMode_ : bool`
- ▣ `batData_ : BatteryManager::BatData`

- `init(void) : void`
- `bind(ledA : gpio::LedController*, ledB : gpio::LedController*, ledC : gpio::LedController*, buttonA : gpio::Button*, buttonB : gpio::Button*, buttonC : gpio::Button*) : void`
- `processEvent(event : PicoEvent*) : bool`
- `startBehaviour(void) : void`
- `getInstance(void) : PicoWatch*`
- `picoWatchTaskHandler(work : struct k_work*) : void`
- `picoWatchTimeoutHandler(timer_id : struct k_timer*) : void`
- `onSecond(calendar : cal::Calendar*) : void`
- `onConnected(lte : LTE*) : void`
- `onDisconnected(lte : LTE*) : void`
- `onError(lte : LTE*) : void`
- `onConnected(mqtt : MQTTController*) : void`
- `onDisconnected(mqtt : MQTTController*) : void`
- `onError(mqtt : MQTTController*, error : int) : void`
- `onReceive(gnss : GnssController*) : void`
- `onConnected(gnss : GnssController*) : void`
- `onDisconnected(gnss : GnssController*) : void`
- `onError(gnss : GnssController*) : void`
- `onInterrupt(batterManager : BatteryManager*) : void`
- `onBatLvlValReady(batterManager : BatteryManager*) : void`
- `onError(batterManager : BatteryManager*, error : int) : void`
- ◇ `pushEvent(PicoEvent* event) : void`
- ◇ `mqttInternalProcessEvent(PicoEvent* event) : void`
- ◇ `sendMqttLocation() : void`
- ◇ `sendMqttTimeDate() : void`
- ◇ `sendMqttMessage_(MQTTController* mqtt, string json, string topic) : void`
- ◇ `startConnectionTimeout_() : void`
- ◇ `onButtonSingleClick(gpio::Button* button) : void`
- ◇ `onButtonDoubleClick(gpio::Button* button) : void`
- ◇ `onButtonTripleClick(gpio::Button* button) : void`
- ◇ `onButtonLongPress(gpio::Button* button) : void`
- ◇ `onButtonDoubleLongPress(gpio::Button* button) : void`
- ◇ `onButtonTripleLongPress(gpio::Button* button) : void`
- ◇ `onButtonVeryLongPress(gpio::Button* button) : void`
- ◇ `onButtonReleased(gpio::Button* button) : void`
- ◇ `button(int index) const : const gpio::Button*`
- ◇ `button(int index) : gpio::Button*`
- ◇ `nextID(uint8_t* pID, uint8_t MAX, uint8_t MIN = 0) : void`
- ◇ `previousID(uint8_t* pID, uint8_t MAX, uint8_t MIN = 0) : void`
- `PicoWatch()`
- `~PicoWatch()`
- `pushDefaultEvent_(delay = 0 : int) : void`
- `_allLedsOff() : void`
- `_allLedsOn() : void`
- `setMotorPosition(uint8_t buttonAction) : void`
- `setGnssAnt(uint8_t buttonAction) : void`
- `disconnect_() : void`
- `mqttConnect_() : void`
- `_ST_SYSTEM_OFF_Action() : void`