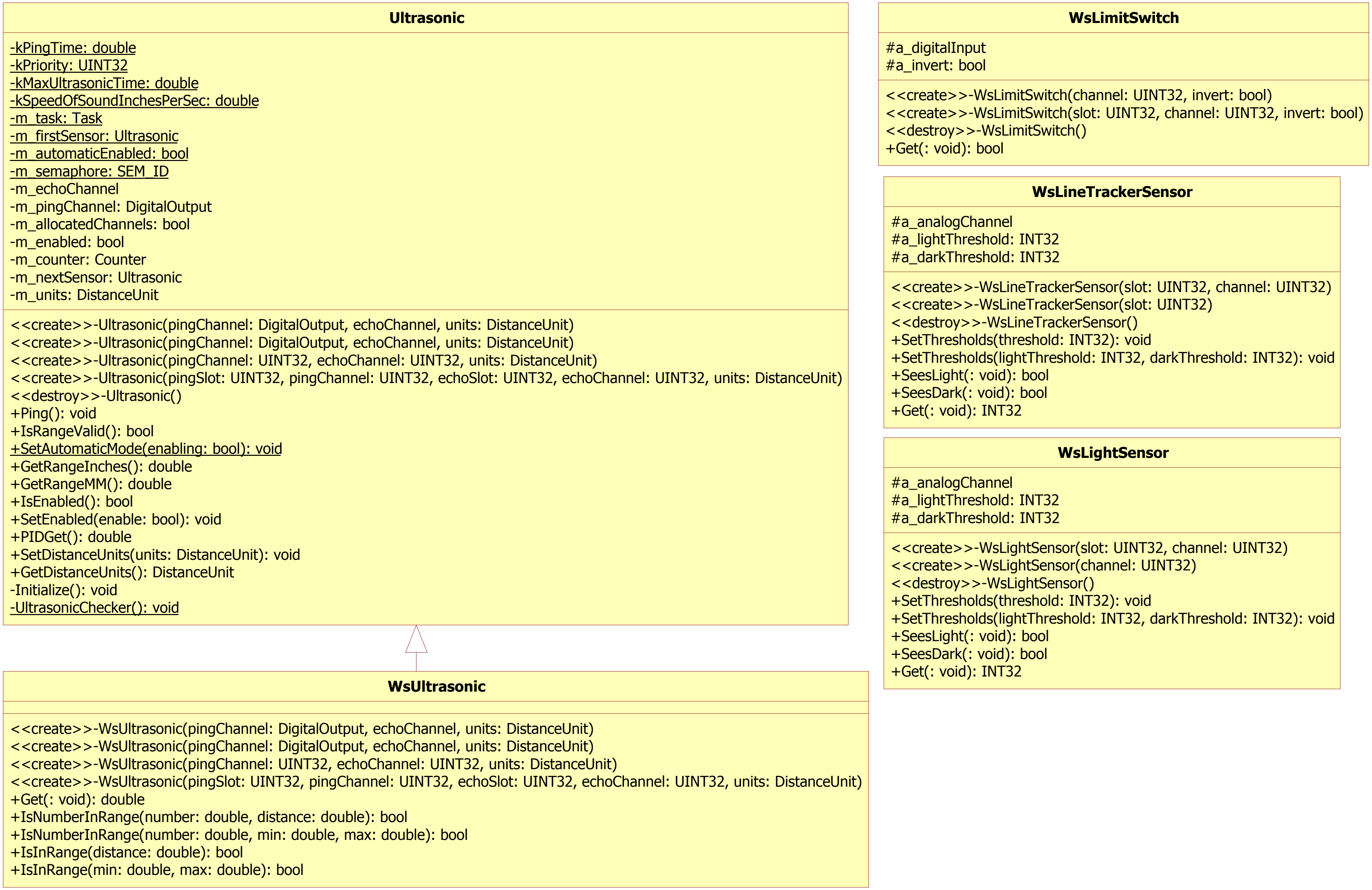

**Ultrasonic**

-kPingTime: double
-kPriority: UINT32
-kMaxUltrasonicTime: double
-kSpeedOfSoundInchesPerSec: double
-m_task: Task
-m_firstSensor: Ultrasonic
-m_automaticEnabled: bool
-m_semaphore: SEM_ID
-m_echoChannel
-m_pingChannel: DigitalOutput
-m_allocatedChannels: bool
-m_enabled: bool
-m_counter: Counter
-m_nextSensor: Ultrasonic
-m_units: DistanceUnit

<<create>>-Ultrasonic(pingChannel: DigitalOutput, echoChannel, units: DistanceUnit)
<<create>>-Ultrasonic(pingChannel: DigitalOutput, echoChannel, units: DistanceUnit)
<<create>>-Ultrasonic(pingChannel: UINT32, echoChannel: UINT32, units: DistanceUnit)
<<create>>-Ultrasonic(pingSlot: UINT32, pingChannel: UINT32, echoSlot: UINT32, echoChannel: UINT32, units: DistanceUnit)
<<destroy>>-Ultrasonic()
+Ping(): void
+IsRangeValid(): bool
+SetAutomaticMode(enabling: bool): void
+GetRangeInches(): double
+GetRangeMM(): double
+IsEnabled(): bool
+SetEnabled(enable: bool): void
+PIDGet(): double
+SetDistanceUnits(units: DistanceUnit): void
+GetDistanceUnits(): DistanceUnit
-Initialize(): void
-UltrasonicChecker(): void

**WsUltrasonic**

<<create>>-WsUltrasonic(pingChannel: DigitalOutput, echoChannel, units: DistanceUnit)
<<create>>-WsUltrasonic(pingChannel: DigitalOutput, echoChannel, units: DistanceUnit)
<<create>>-WsUltrasonic(pingChannel: UINT32, echoChannel: UINT32, units: DistanceUnit)
<<create>>-WsUltrasonic(pingSlot: UINT32, pingChannel: UINT32, echoSlot: UINT32, echoChannel: UINT32, units: DistanceUnit)
+Get(: void): double
+IsNumberInRange(number: double, distance: double): bool
+IsNumberInRange(number: double, min: double, max: double): bool
+IsInRange(distance: double): bool
+IsInRange(min: double, max: double): bool

**WsLimitSwitch**

#a_digitalInput
#a_invert: bool

<<create>>-WsLimitSwitch(channel: UINT32, invert: bool)
<<create>>-WsLimitSwitch(slot: UINT32, channel: UINT32, invert: bool)
<<destroy>>-WsLimitSwitch()
+Get(: void): bool

**WsLineTrackerSensor**

#a_analogChannel
#a_lightThreshold: INT32
#a_darkThreshold: INT32

<<create>>-WsLineTrackerSensor(slot: UINT32, channel: UINT32)
<<create>>-WsLineTrackerSensor(slot: UINT32)
<<destroy>>-WsLineTrackerSensor()
+SetThresholds(threshold: INT32): void
+SetThresholds(lightThreshold: INT32, darkThreshold: INT32): void
+SeesLight(: void): bool
+SeesDark(: void): bool
+Get(: void): INT32

**WsLightSensor**

#a_analogChannel
#a_lightThreshold: INT32
#a_darkThreshold: INT32

<<create>>-WsLightSensor(slot: UINT32, channel: UINT32)
<<create>>-WsLightSensor(channel: UINT32)
<<destroy>>-WsLightSensor()
+SetThresholds(threshold: INT32): void
+SetThresholds(lightThreshold: INT32, darkThreshold: INT32): void
+SeesLight(: void): bool
+SeesDark(: void): bool
+Get(: void): INT32