smacc::ISmaccClient # stateMachine # components + ISmaccClient() + ~ISmaccClient() + initialize() + setStateMachine() + getName() + postEvent() + postEvent() + getComponent() + configureEventSourceTypes() smacc::client bases ::SmaccSubscriberClient < MessageType > + topicName + queueSize + onFirstMessageReceived + onMessageReceived_ + postMessageEvent + postInitialMessageEvent # nh_ - sub - firstMessage_ initialized_ + SmaccSubscriberClient() + ~SmaccSubscriberClient() + onMessageReceived() + onFirstMessageReceived() + configureEventSourceTypes() + initialize() messageCallback() multirole sensor client ::ClMultiroleSensor< Message Type > + onMessageTimeout_ + postTimeoutMessageEvent + timeout timeoutTimer_ - initialized_ + ClMultiroleSensor() + onMessageTimeout() + configureEventSourceTypes() + initialize() # resetTimer() timeoutCallback() < sensor_msgs::LaserScan > < sensor_msgs::Temperature > multirole_sensor_client multirole_sensor_client ::ClMultiroleSensor< sensor ::CIMultiroleSensor< sensor _msgs::LaserScan > _msgs::Temperature > + onMessageTimeout + onMessageTimeout + postTimeoutMessageEvent + postTimeoutMessageEvent + timeout + timeout - timeoutTimer - timeoutTimer - initialized initialized + ClMultiroleSensor() + ClMultiroleSensor() + onMessageTimeout() + onMessageTimeout() + configureEventSourceTypes() + configureEventSourceTypes() + initialize() + initialize() resetTimer() resetTimer() - timeoutCallback() timeoutCallback() sm dance bot 2::cl sm_dance_bot::cl_temperature lidar::ClLaserSensor sensor::ClTemperatureSensor + ClLaserSensor() + ClTemperatureSensor()

sm_dance_bot::cl_lidar

::ClLaserSensor

+ ClLaserSensor()