Truck

- + dummy_client_IDs_vec_mutex: pthread_mutex_t
- + self: TruckMetadata
- + t_communication: pthread_t
- + t_controller: pthread_t
- + t_decryptor: pthread_t+ t_interface: pthread_t
- + truck_communication: CommsModule = CommsModule(0,1...
- + truck controller: controller = controller(0, &self)
- + truck_decryptor: Decryptor = Decryptor(&self)
- + run(): void

CommsModule

- + client_IDs: std::vector<int>
- client_IDs_vec_mutex: pthread_mutex_t
- clientSocket: SOCKET = {}
- ID: int
- rx_messages: std::vector<Message>
- rx_vec_mutex: pthread_mutex_t
- + self_truck: TruckMetadata*
- serverAddr: sockaddr_in = {}
- timeout: timeval = {}tx_messages: std::vector<Message>
- tx_vec_mutex: pthread_mutex_t
- + add_tx_message_to_buffer(Message): int
- + CommsModule(int, long, TruckMetadata*)
- + connect_to_Server(): int
- + get_connected_client_IDs(): std::vector<int>
- + get_last_rx_message_from_buffer(bool): std::optional<Message>
- + get_length_of_rx_buffer(): int
- + get_rx_message_by_index_from_buffer(int, bool): std::optional<Message>
- + initialize(std::string&, u_short): int
- + print_rx_messages_from_buffer(): int
- + receive_rxBuffer(): int
- receive_string(std::string&): int
- + <u>run(void*): void*</u>
- + run_thread(): void*
- send_message(Message): int
- send_string(std::string&): int
- + send_txBuffer(): int

SocketClientID

- + clientSocket: SOCKET
- + ID: int

SocketClientID

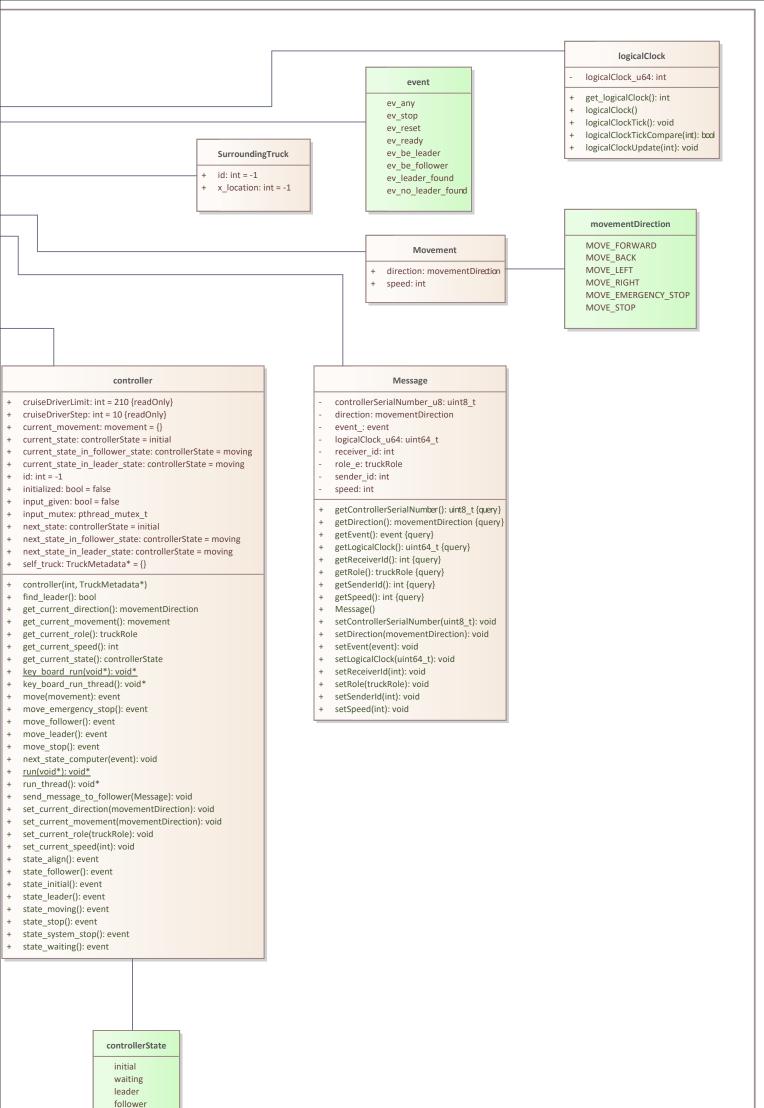
- + clientSocket: SOCKET
- ID: int

MessageParser

- + <u>directionToString(movementDirection): std::string</u>
- + fromJSON(std::string&): Message
- + <u>stringToDirection(std::string&): movementDirection</u>
- + <u>stringToTruckRole(std::string&): truckRole</u>
- + toJSON(Message&): std::string
- + truckRoleToString(truckRole): std::string

TruckMetadata

- + client_IDs_vec_mutex: pthread_mutex_t*
- + event_handler: event = ev_any
- + movement_leader: std::vector<movement>
- movement_leader_vec_mutex: pthread_mutex_t
- + pending_send_message: std::vector<Message>
- + received_message: std::vector<Message>
- + received_message_vector_guard: std::mutex
- + received_message_vector_mutex: pthread_mutex_t
- role: truckRole = NOT_SET
- + send_message_vector_mutex: pthread_mutex_t
- + send_messsage_vector_guard: std::mutex
- + surrounding_truck: std::vector<SurroundingTruck>
- + surrounding_truck_IDs: std::vector<int>*
- + truck_id: int = -1
- truck_leader_id: int = -1
- + truck_logical_clock: logicalClock
- + watchdog: time_t



moving



aligning stop system_stop