Open source multicopter library

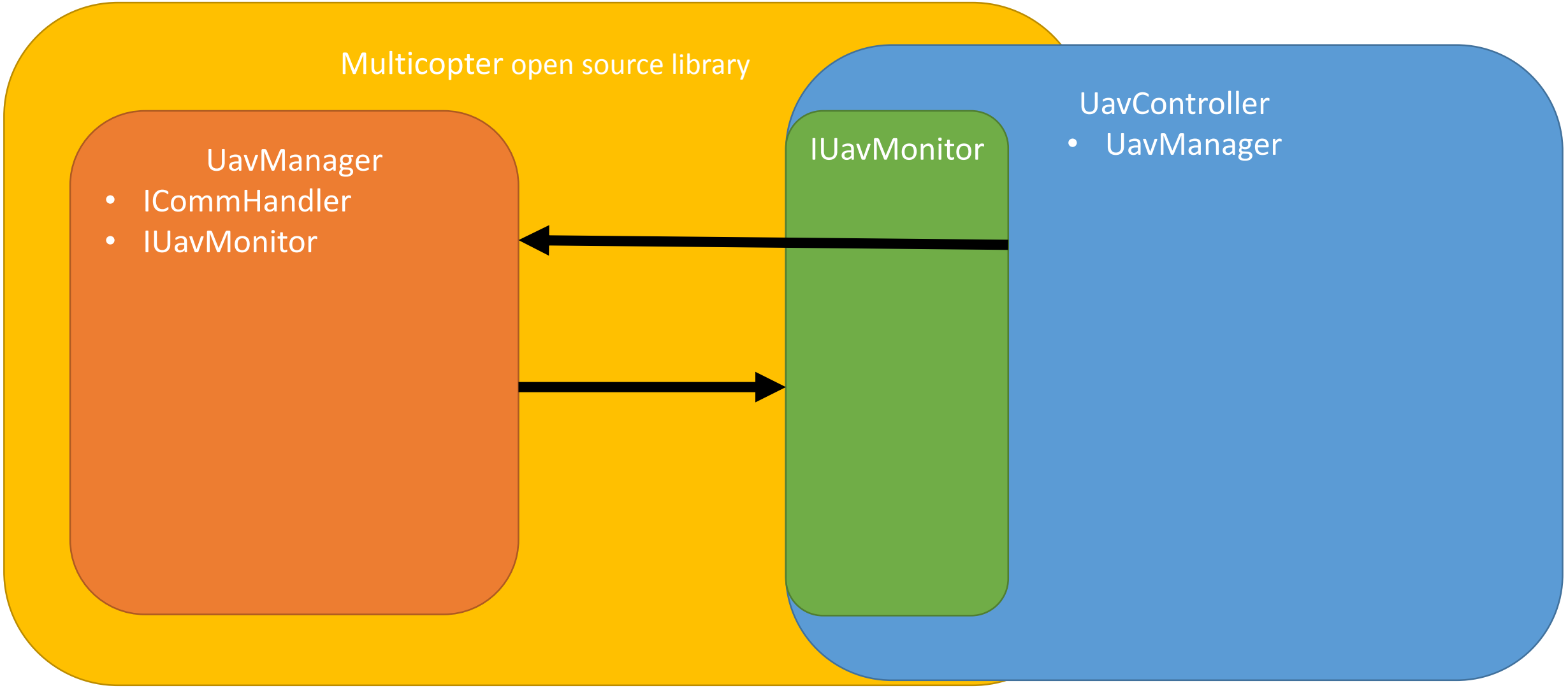
Data structure

- multicopter *(git repository root)*
  - cpp
    - include
      - communication
      - common
      - user_app
    - source
      - communication
      - common
      - user_app
  - java
    - actions
    - data
    - events
  - python

C++ classes

- communication
  - IMessage
  - ISignalPayloadMessage
  - SignalData
  - ControlData
  - DebugData
  - AutopilotData
  - SensorsData
  - CalibrationSettings
  - ControlSettings
  - RouteContainer
  - CommDispatcher
  - *ICommInterface*
  - *ICommHandler*
- common
  - MathCore
    - Vect(2,3,4)D<_Tp>
    - Mat(2,3,4)D<_Tp>
    - Vector<_Tp>
    - Matrix<_Tp>
  - Location
  - Waypoint
  - ImuData
  - GpsData
- user_app
  - *IAppCommHandler*
  - UavManager
  - *IUavMonitor*
  - UavEvent
  - UserUavEvent
  - Exception

C++ user application model



Running communications loop

- ***ICommInterface*** handles communication,
  - *send* – sends bytes array over specific interface, returns true if successfull
  - *getData* – returns true if any data is avalible and then one byte (unsigned char) value is set under pointer passed as method argument
- *runAppLoop* and *runExternalSensorsLogger* methods shall be called from another thread, that threads end when connetion will be shutdown after user disconnect command
- Thread hold is handled by ***IAppCommHandler***::*holdThread* and this implementation shall be compatibile with thread framework that is used, example:
  - std::thread needs std::this_thread::*sleep_for*
  - WinApi: *_beginethread* needs Winapi: *Sleep,* etc.

Interfaces

- ***ICommInterface*** – „phiscal” layer communication issues
  - *bool send(const unsigned char* const data)*
  - *bool getData(char* data)*
- ***IAppCommHandler*** – time handling issues
  - *void holdThread(const unsigned miliseconds)*
  - *void restartTimer(void)*
  - *unsigned getTimerValue(void)*
- ***IUavMonitor*** – receiving events from UAV
  - *void notifyUavEvent(const UavEvent* const event)*
  - *void notifyDataReceived(const IMessage* const data)*
  - *void notifyPingUpadted(const float miliseconds)*

UavManager

- runAppLoop(ICommInterface* commInterface)
- runExternalSensorsLogger(ICommInterface* commInterface)
- preformAction(const UavAction& uavAction)
- preformActionUpload(const ControlSettings* const controlSettings)
- preformActionUpload(const RouteContainer* const routeContainer)
- setExternalEvent(const ExternalEvent& externalEvent)
- updateControlData(const ControlData& controlData)
- updateAutopilotData(const AutopilotData& autopilotData)

Java

Java

- Classes are in packadge „**com.java.multicopter.xxx**”

Comm objects structure

- CommInterface – CommDispatcher – CommHandler - UavManager