MAVConn Tutorial

Computer Vision Lab Tutorial

27 September 2011

Lorenz Meier, Friedrich Fraundorfer, Kevin Koeser

Schedule

pixhawk

http://pixhawk.ethz.ch

- Today: Introduction to MAVConn
- Tomorrow morning: MAVConn with MAVLink v1.0.0
- Tomorrow morning: MAVConn exercise / example provided, full slide set provided
- Thursday morning: Resubmission of project plans
- Friday: Vicon and Camera calibration hands-on tutorial
- Early next week: Project plan reviews

MAVLink Protocol (1/4)

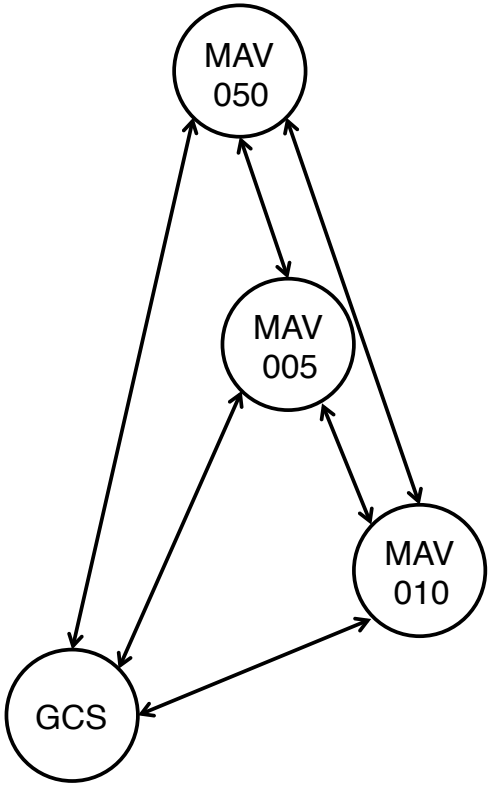
pixhawk

http://pixhawk.ethz.ch

MAVLINK

MICRO AIR VEHICLE COMMUNICATION PROTOCOL

- Efficient Protocol for small autonomous Systems
- Support for 255 systems
- ***BROADCAST*** – only sender id in packet
- ***LOSS*** – Retransmission implemented on application level, not inprotocol
- ***STREAM MODEL*** – Data is sent in regular intervals, ***forward error correction-like approach***

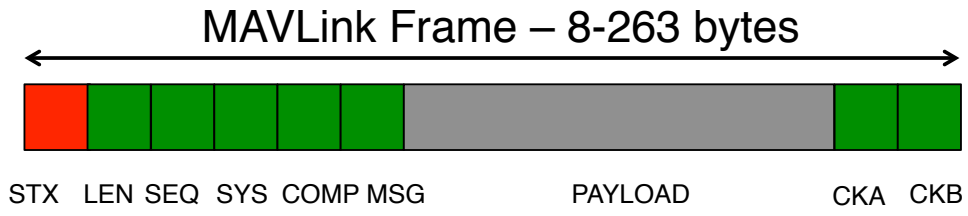


MAVLink Protocol (2/4)

pixhawk
http://pixhawk.ethz.ch

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

- Packet Anatomy: 4 bytes header, 2 bytes checksum
- 255 bytes maximum payload length
- For packetized links (Ethernet, Wifi), and large data: Send five packets at once (1275 bytes, below Ethernet MTU)



MAVLink Protocol (3/4)

pixhawk

http://pixhawk.ethz.ch

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

- Protocol Definition: XML file
- Generator: Generates C and Python Code to serialize messages

```
<?xml version='1.0'?>
<mavlink>
     <version>3</version>
     <enums>
          <enum name="MAV_AUTOPILOT">
               <entry value="0" name="MAV_AUTOPILOT_GENERIC">
               </entry>
               <entry value="1" name="MAV_AUTOPILOT_PIXHAWK">
               </entry>
          </enum>
     <messages>
          <message id="0" name="HEARTBEAT">
               <field type="uint8_t" name="type">Type of the MAV</field>
               <field type="uint8_t" name="autopilot">Autopilot type</field>
               <field type="uint8_t" name="base_mode">System mode bitfield</field>
               <field type="uint32_t" name="custom_mode">Navigation mode bitfield</field>
               <field type="uint8_t" name="system_status">System status flag</field>
               <field type="uint8_t_mavlink_version" name="mavlink_version">MAVLink version</field>
          </message>
     </messages>
</mavlink>
```

MAVLink Protocol (4/4)

pixhawk

http://pixhawk.ethz.ch

- C Implementation: ***Header only Library***
- Very efficient implementation

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

```
#define MAVLINK_MSG_ID_HEARTBEAT 0

typedef struct __mavlink_heartbeat_t
{
 uint32_t custom_mode; ///< Navigation mode bitfield
 uint8_t type; ///< Type of the MAV
 uint8_t autopilot; ///< Autopilot type / class
 uint8_t base_mode; ///< System mode bitfield
 uint8_t system_status; ///< System status flag
 uint8_t mavlink_version; ///< MAVLink version
} mavlink_heartbeat_t;


static inline uint16_t mavlink_msg_heartbeat_pack(uint8_t system_id, uint8_t component_id, mavlink_message_t* msg,
                                 uint8_t type, uint8_t autopilot, uint8_t base_mode, uint32_t custom_mode, uint8_t system_status)
{
    mavlink_heartbeat_t packet;
    packet.custom_mode = custom_mode;
    packet.type = type;
    packet.autopilot = autopilot;
    packet.base_mode = base_mode;
    packet.system_status = system_status;
    packet.mavlink_version = 3;

        memcpy(_MAV_PAYLOAD(msg), &packet, 9);

    msg->msgid = MAVLINK_MSG_ID_HEARTBEAT;
    return mavlink_finalize_message(msg, system_id, component_id, 9, 50);
}
```

MAVConn Architecture 1/2

pixhawk

http://pixhawk.ethz.ch

MAVCONN

MICRO AIR VEHICLE ONBOARD IPC MIDDLEWARE

- ***EVERYTHING*** is broadcast
- Acknowledgements are on application level (e.g. parameter read/write)
- Positive default acknowledgement (heartbeat)
- Forward Error Correction (data streams at 1.25x – 2x the rate needed)
- Wrong data / fault: Drop and Continue

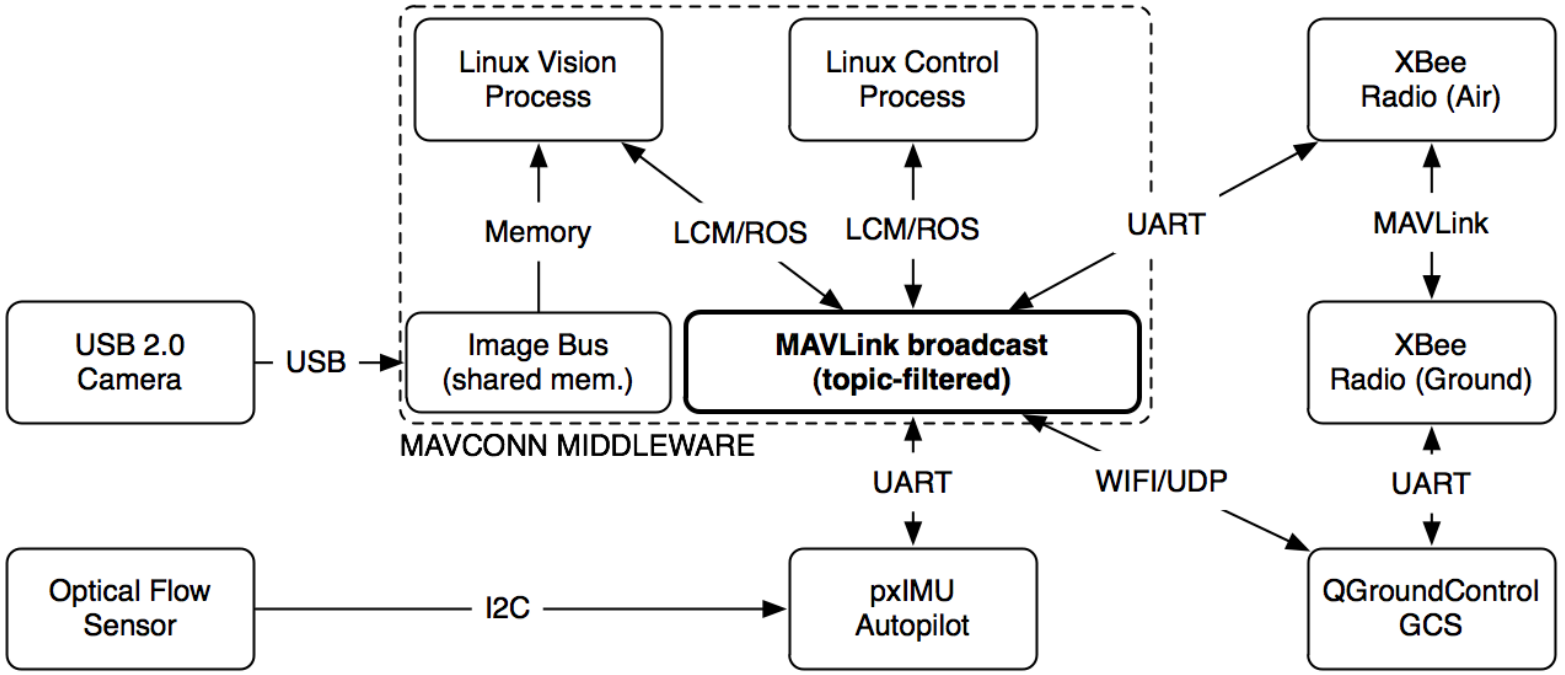
MAVConn Architecture (2/2)

pixhawk

http://pixhawk.ethz.ch

MAVCONN

MICRO AIR VEHICLE ONBOARD IPC MIDDLEWARE



System Control

pixhawk

http://pixhawk.ethz.ch

- px_system_control
- Process to shutdown / control the whole computer
- Won't shutdown the PC as long as its not run as super-user
- Can emit heartbeat

UDP Bridge

pixhawk

http://pixhawk.ethz.ch

- Forwards MAVLink messages from internal bus to external communication partner
- Start one line per communication partner

```
mavconn-bridge-udp –r <ip or hostname>
```

Serial Bridge

pixhawk

http://pixhawk.ethz.ch

- Forwareds MAVLink messages from LCM bus to a serial / UART line
- Can be used to interface an autopilot or radio modem

```
mavconn-bridge-serial
```

Camera Interface

pixhawk

http://pixhawk.ethz.ch

- Reads one or two cameras synchronized
- Supports PointGrey, MatrixVision and Kinect
- Supports fixed exposure and auto-exposure
- Provides synchronized data
- Puts images and meta data onto shared memory
- Reads first camera in system

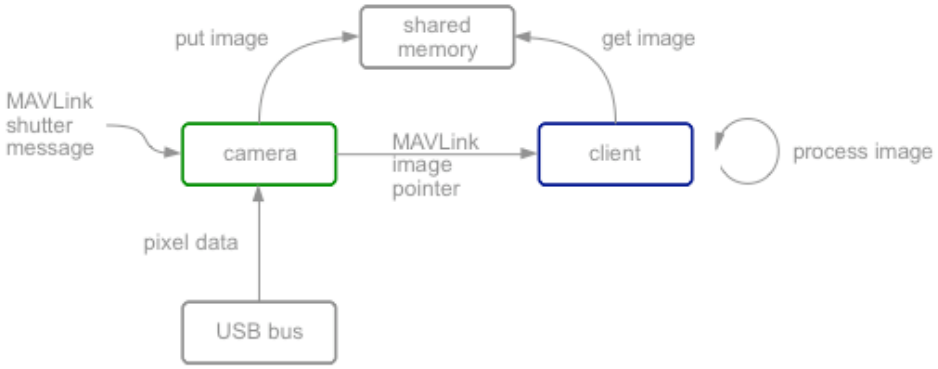
```
mavconn-camera
```

Shared Memory

pixhawk

http://pixhawk.ethz.ch

- In a vision-enabled MAV typically several algorithms work on the same image
- Camera can't be "owned" by a single process
- Efficient and fast image-sharing required



Why Synchronization? (1/2)

http://pixhawk.ethz.ch

Position of red time instant:
$P(t) = P_{stereo}(t-1) + (P_{odometry}(t) - P_{odometry}(t-1))$

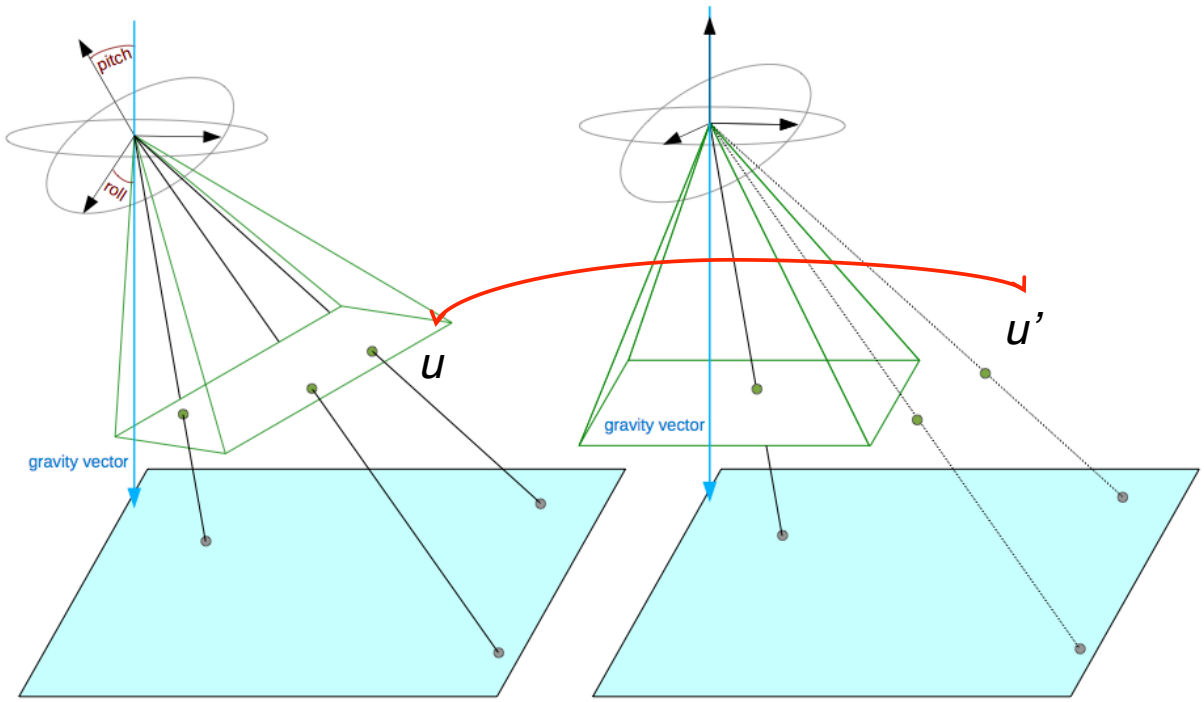
Stereo Pose estimate at (t-1)

Why Synchronization? (2/2)

pixhawk

http://pixhawk.ethz.ch

$$u' = R_{\phi\theta} K^{-1} u$$



Logging

pixhawk

http://pixhawk.ethz.ch

- Log Line: uint64_t timestamp + 263 bytes for packet
- Allows exact mission replays
- Start px_imagecapture , then send logging start / stop commands via QGroundControl
- Will log to ~/dataset_capture
  - <timestamp>.mavlink – MAVLink logfile, can replayed with log replay or QGroundControl
  - Left: Left camera image
  - Right: Right camera image

Log Replay

pixhawk

http://pixhawk.ethz.ch

- Start px_replay with these arguments
  - "--logfile" with the correct Path to the .mavlink file
  - "--orientation" (forward|downward), depending on setup
- Start px_view to see the replayed images
- Start px_mavlink_bridge_udp to route data to ground control unit
- Start QGroundControl to view data

QGroundControl

pixhawk

http://pixhawk.ethz.ch

Assignment

pixhawk

http://pixhawk.ethz.ch

- Assignment online: Wednesday morning (tomorrow)
- Assignment due: Monday morning (October 24)
- Required effort: 5 mins programming, 30 mins setup
- Assignment task:
  - Setup your notebook with Ubuntu and MAVConn
  - Implement example process according to assignment
  - Process provided logfile, hand in textfile with console output and stored images