Idea report

Android MISL Control

Android Practical Summer Term Course (SS15)

Fent, Philipp

Sautermeister, Benjamin

05/11/2015

Table of contents

1. Introduction

This idea report provides a summary of ideas, goals and workload of the project “Android MISL Control”, developed for the Android Practical course in the summer semester of 2015. The described project will be realized in an international cooperation between Texas A&M University (TAMU) and Technical University of Munich (TUM).

1.1 Problem

The department of Electrical Engineering of TAMU, in cooperation with the National Aeronautics and Space Administration's (NASA) Johnson Space Center (JSC)[1], developed a Modular Integrated Stackable Layer (MISL) platform. This platform allows building a modular, extensible and space-qualified hardware system, where you can stack different kind of general or specific purpose boards on top of each other.

To showcase the potential of MISL, TAMU introduced the Articulated Suspension Exploratory Platform (ASEP), a remote controlled rover for hazardous terrain, powered by a MISL microcontroller. ASEP was developed by TAMU students in a Capstone Project in 2014[2] and provides a Windows executable Graphical User Interface (GUI) application. This program allows controlling the robot and displaying its current status information. But unfortunately, it cannot be executed on modern mobile devices such as Android devices.

1.2 Project goal

The main goal of Android MISL Control is to provide an Android application, which mirrors the functionality of the current Windows GUI. This includes the following aspects:

- Control the robot (Send direction commands)
- Display information transmitted by ASEP
  - Situation (Pitch / Roll Yaw)
  - Position (Latitude, Longitude)
  - Graphical visualization of ASEPs situation, positions and current steering

[1] More information: *http://engineering.tamu.edu/news/2013/08/23/texas-am-engineering-students-to-collaborate-with-nasa*

[2] More information: *http://asee-gsw.tulane.edu/pdf/the-modular-integrated-stackable-layers-system-a-nasa-development-partnership.pdf*

Furthermore, new interaction capabilities will be enabled due to the integrated sensors of an Android phone or tablet. Additionally, all the included components of the Android Software Development Kit (SDK) allow new forms of data representation and visualization.

1.3 Project team

The project development team consists of three students in total, which will all report their progress directly to Dr. Morgan of TAMU. Additionally, all German students should refer to the advisor Nils Kannengießer with all questions regarding Android software development.

First name	Last name	Institution	Function
Philipp	Fent	TUM	Software Engineering
Benjamin, BSc.	Sautermeister	TUM	Software Engineering
TBA	TBA	TAMU	Electrical Engineering / Software User Acceptance Testing

Table 1 - Project development team

For the sake of completeness, the following table lists all project stakeholders including their contact information:

First name	Last name	Institution	Email	Function
Joseph, A., Dr.	Morgan	TAMU	*morganj@entc.tamu.edu*	Advisor
Nils, MSc.	Kannengießer	TUM	*nils.kannengiesser@tum.de*	Advisor
Philipp	Fent	TUM	*fent@in.tum.de*	Student
Benjamin, BSc.	Sautermeister	TUM	*benjamin.sautermeister@tum.de*	Student
TBA	TBA	TAMU	TBA	Student

Table 2 - Contact information

2. Status quo

The following section describes the existing software application, which will be ported to the Android operating system during the project. Furthermore, an insight into the existing communication protocols and temporary camera integration will be given.

2.1 Windows GUI

LabVIEW[3] is a system-design and development environment, particularly popular with electrical engineering and industrial automation. Software applications in LabVIEW are programmed in a graphical programming language called G and the current implementation of the GUI relies on a quite complex graphical control flow.

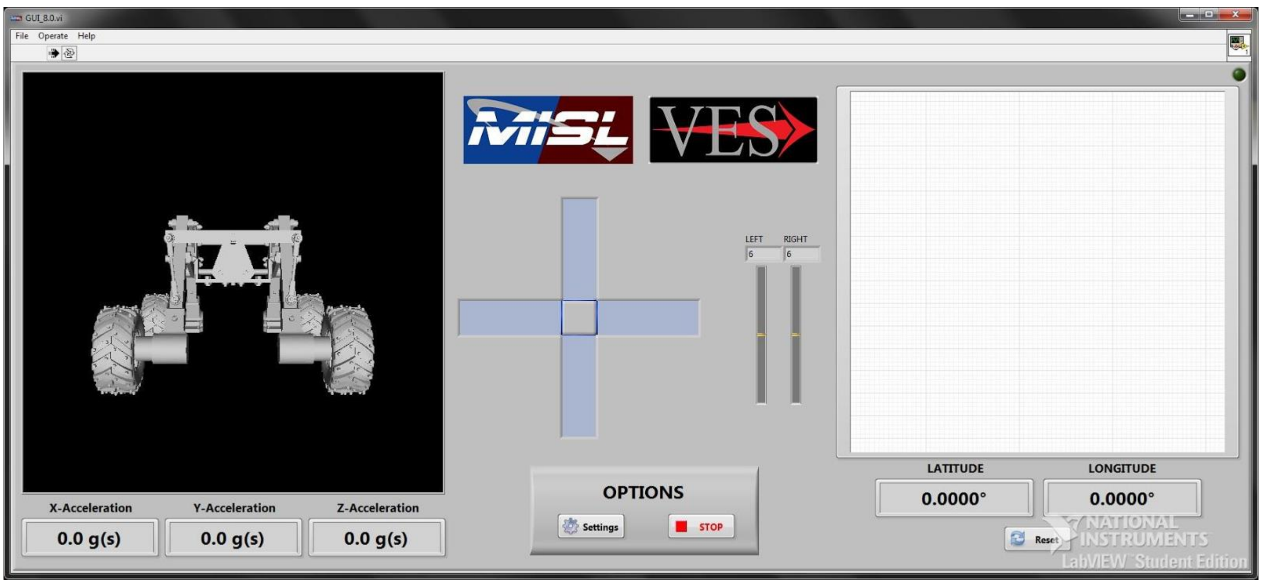

Figure 1 - LabView Windows GUI application

The GUI consists of three sections, which are illustrated in Figure 1 :

- Left: Display of AHRS[4] data with visualization of a 3D model.
- Center: Display of current control information to be sent to the robot
- Right: Plot of ASEPs latest known GPS positions relative to its starting position

2.2 Control mechanism

The current control mechanism consists of an electronic joystick, ordinarily used in flight simulations. The joystick is connected to the PC via USB running the GUI. Then the joysticks X- and Y- manipulation is

[3] More information: http://www.ni.com/labview/d/
[4] Acronym for: Attitude and Heading Reference System

translated into CH1 and CH2 commands[5] and sent to the MISL stack powering the robot. This results in a quite intuitive steering with the additional possibility to control ASEP quite precisely. This could not be realized with buttons of a keyboard, which only allow binary-like commands.

2.3 Communication protocol

The MISL stack sets up a WiFi-direct[6] Network and an UDP Server (IP: 192.168.16.254, Port: 30190). The current communication protocol between the GUI and MISL is described in the ASEP projects final report (p. 78ff) in full detail.

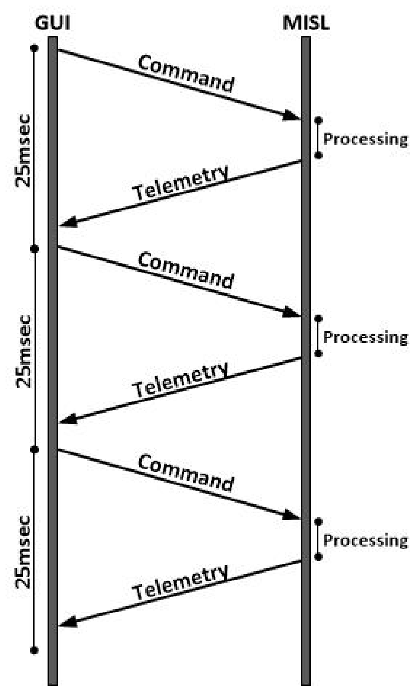

Figure 2 - Communication between GUI application and MISL

Figure 2 shows the individual communication packets between the GUI application and MISL. The application sends command packets, containing the steering information (compare Figure 3 and Figure 4), periodically in 25ms intervals.

	Packet Version	Size	Receive SeqCnt	Command	CH1 Cmd	CH2 Cmd	Checksum	Padding
Command	0x02 0x03	0x00 0x40	0x75 0x65	0x75 0x65	0x00 0x00	0x00 0x00	0x00 0X00	25 * 0x0000

Figure 3 - Command packet

[5] Refer to *Final_Presentation_v1.2.pptx* of the VES Capstone project, slide 90 for detailed source code
[6] Wi-Fi standard enabling devices to easily connect with each other without requiring a wireless access point

These packets trigger a telemetry packet response of ASEPs microcontroller. Each telemetry packet contains the sensor information measured. All sensor information are transmitted in single precision (binary32) IEEE floating point values.

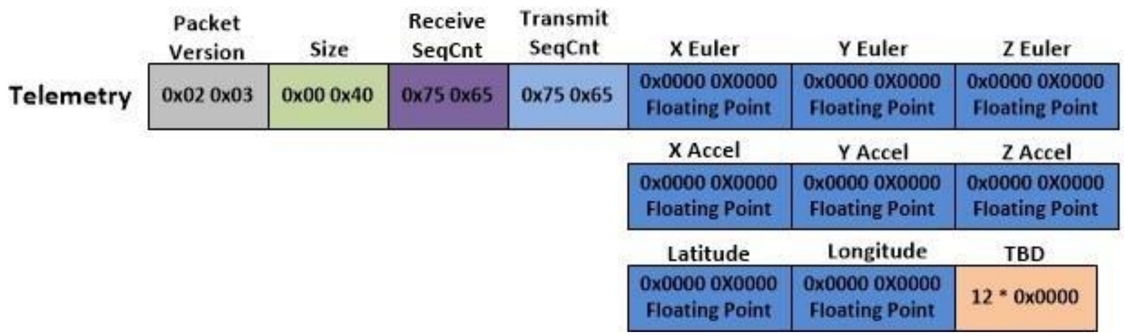

Figure 4 - Telemetry packet

The following list gives an overview of the sensor information of each telemetry packet:

- Gyroscopic information in Euler angles
- Acceleration information for each axis
- GPS coordinates in latitude and longitude

2.4 Camera integration

The current implementation of transmitting a camera stream from the ASEP to the controlling computer is rather provisional. The video stream is currently sent from the camera separately to an external camera receiver program, which is not integrated in the Windows application. To receive the camera signal, there is also the need for an external antenna plugged into the computer. Since this is not a complete solution, there is probably no easy way to integrate this camera without the external antenna.

3. Requirements

This chapter gives an overview of all identified requirements which are in scope of the project. Additional or changes in requirements could arise during the development advances.

ID	1	**Type**	Functional
Priority	MUST	**Originator**	Dr. Morgan
Title	Wireless communication between MISL and Android application		
Description	A wireless connection between the MSIL controller of the ASEP robot and the Android application has to be established. The Android application has to implement the communication protocols which are implemented in the current MSIL and LabView application.		

ID	2	**Type**	Functional
Priority	OPTIONAL	**Originator**	Dr. Morgan
Title	Communication configuration guidance		
Description	The user should be assisted on how to establish the WiFi-direct connection to ASEP.		

ID	3	**Type**	Functional
Priority	MUST	**Originator**	Dr. Morgan
Title	Controlling the ASEP robot		
Description	The Android application has to be able to send control commands to the ASEP robot; hence it can be remotely operated. The control mechanism can be both, using a virtual joystick or a sensor-based approach. Also a hybrid implementation could be possible.		

ID	4	**Type**	Functional
Priority	MUST	**Originator**	Dr. Morgan
Title	Display of gyroscope status information		
Description	The current situation of ASEP based on its sensors should be displayed as text.		

ID	5	**Type**	Functional
Priority	OPTIONAL	**Originator**	Dr. Morgan
Title	Display of the 3D ASEP model		
Description	Based on the current situation of the ASEP, a 3D model of the robot should be rendered in OpenGL. The transformation of that model reflects the latest information of the gyroscope sensor.		

ID	6	**Type**	Functional
Priority	MUST	**Originator**	Dr. Morgan
Title	Display of location information		
Description	The current location of ASEP based on its integrated GPS sensor should be displayed as text. Optionally, the location of the robot can be displayed in a Google Maps with a drawn route of its driven path.		

ID	7	**Type**	Constraint
Priority	MUST	**Originator**	Dr. Morgan
Title	Samsung Galaxy Tab 10.1 and Android 4.0.4 as target platform		
Description	The provided Samsung tablet deals as the main target platform. Therefore, all optimizations and testing should be performed on that device. In case of there is a particularly need of a higher Android version than 4.0.4 (API Level 15), an update is easily conceivable with permissions by Dr. Morgan.		

ID	8	**Type**	Functional
Priority	SHOULD	**Originator**	Dr. Morgan
Title	Display of the camera's video stream		
Description	The Android application should display the video stream provided by the integrated camera. In case there is a need of a different camera, Texas A&M University could provide another model based on our needs.		

4. Realization

This chapter describes important aspects regarding the realization of the international project. It includes project uncertainties, project collaboration and a detailed project plan.

4.1 Uncertainties

Possible risks and uncertainties of the project were identified and listed in the following table:

ID	Description
1	There is no spare ASEP robot for testing software, which could be shipped to Germany. This means, acceptance testing is quite hard and needs to be done in the US, with the possibility of video conferencing.
2	Building a MISL stack in ASEP configuration is currently planned, but there is no ETA for Germany yet. This means testing the wireless communication is quite hard and probably needs to be done with another device simulating in the meantime.
3	The third team member from TAMU is not yet announced.
4	The current camera implementation cannot be transferred to an Android implementation. There is a need to research possible alternative cameras and test them for Android compatibility.
5	There is a 3D model of ASEP available to be displayed, but libraries to display the STL File format needs to be researched. The display of the 3D model could also be achieved by converting the model to a more mobile friendly format.

Table 3 - Possible project risks and uncertainties

4.2 Collaboration

Programming collaboration is based on the distributed version control system GIT to allow a global development process in this international project. As a collaboration platform, GitHub is used, because it offers free repositories for open source software development. Moreover, it is the market leader for online programming collaboration.

The project is located under the following link: https://github.com/pfent/Android-MISL-Control

4.3 Project plan

This chapter summarizes the activities, milestones and deadlines of the project. All these information is given in the form of a work breakdown structure in Figure 5.

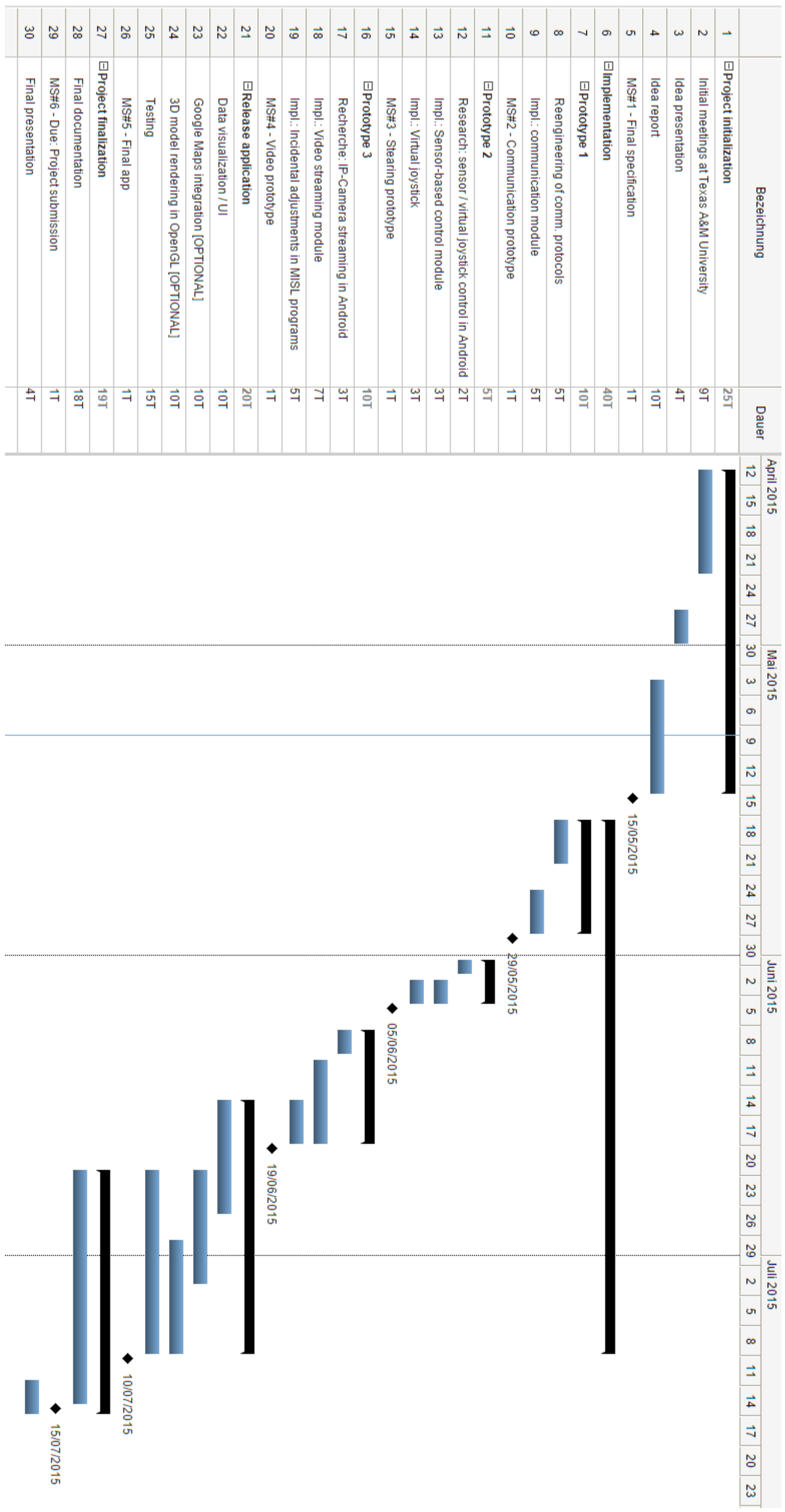

Figure 5 - Project plan (GANTT)

Before starting to implement the final Android application, a total of three prototype applications will be developed. Each application acts as a proof of concept for the main functionality with a clear focus on functionality that comes with a rudimentary user interface only. Subsequently, all these efforts will flow together into the final application.