Project Name	Updating Flash Boot Loader
Online team meeting	https://tu-berlin.zoom.us/j/66771292215?pwd=YUlZRmZoRTFyZkFLelVJQUhFMElzZz09
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2024ss07-updating-flash-boot-loader
GitHub feature board	https://github.com/orgs/amosproj/projects/61
GitHub impediments backlog	https://github.com/orgs/amosproj/projects/66
Team T-shirt (white)	https://www.shirtinator.de/s/DT8swsrfTzyO5puA44dbDw
Team T-shirt (black)	https://www.shirtinator.de/s/lBlKPwM8STWfVG6CVlJbLQ
Additional materials	...
Team maling list	oss-amos-proj7@lists.fau.de
Links	
Happiness Index	https://happy-amos.appspot.com/
Capabilities Timeline	Capabilities Timeline (by Week)
Capabilities Timeline Explained	Capabilities Timeline Explained

Last Name	First Name	GitHub User Name	Email Address
Wilms	Leon	wlklsn	leonwilms.wk@gmail.com
Rodriguez Schmidt	Sebastian Alberto	sebas-rodri	r99@melao.de
Freund	Andreas	AndreasFreund	andi.freund@fau.de
Roy	Paul	PaulRoy1	paul.roy@fau.de
Pilarczyk	Wiktor	wiklam	wiktorpilar99@gmail.com
Ehrl	Dorothea	ibexq	dorothea.ehrl@fau.de
Gtari	Nahrain	Nahrain1	nahraingtari1@gmail.com
Bauer	Michael	MikeFAU	mike.bauer@fau.de
Soni	Rahil	Rahilsoni007	raj.soni@fau.de

#	Meeting Day	Product Owners	Software Developer	Release Manager	Scrum Master	Comment
1	2024-04-17	Andreas Freund, Nahrain Gtari	Everyone else	n/a	COACH student	
2	2024-04-24	Andreas Freund, Nahrain Gtari	Everyone else	n/a	COACH student	
3	2024-05-01	Andreas Freund, Nahrain Gtari	Everyone else	n/a	COACH student	No regular project meeting, public holiday
4	2024-05-08	Andreas Freund, Nahrain Gtari	Everyone else	Dorothea Ehrl	COACH student	
5	2024-05-15	Andreas Freund, Nahrain Gtari	Everyone else	Wiktor Pilarczyk	COACH student	Build process review
6	2024-05-22	Andreas Freund, Nahrain Gtari	Everyone else	Leon Wilms	COACH student	
7	2024-05-29	Andreas Freund, Nahrain Gtari	Everyone else	Michael Bauer	COACH student	
8	2024-06-05	Andreas Freund, Nahrain Gtari	Everyone else	Paul Roy	COACH student	Mid-term due
9	2024-06-12	Andreas Freund, Nahrain Gtari	Everyone else	Sebastian Rodriguez	COACH student	
10	2024-06-19	Andreas Freund, Nahrain Gtari	Everyone else	Dorothea Ehrl	COACH student	
11	2024-06-26	Andreas Freund, Nahrain Gtari	Everyone else	Wiktor Pilarczyk	COACH student	
12	2024-07-03	Andreas Freund, Nahrain Gtari	Everyone else	Michael Bauer	COACH student	
13	2024-07-10	Andreas Freund, Nahrain Gtari	Everyone else	Leon Wilms	COACH student	
14	2024-07-17	Andreas Freund, Nahrain Gtari	Everyone else	Paul Roy	COACH student	Demo day!
15	2024-07-24	Andreas Freund, Nahrain Gtari	Everyone else	Sebastian Rodriguez	COACH student	Retrospective
Product owners, software developers, and Scurm Master are set and ideally don't change over time; the critical part is the Release Manager role you need to define here						

Goals	Having a good time as a team and learning about technology
	Working product that meets requirements
Meeting norms	Keep Agenda. Additions to Agenda need to be communicated in Slack beforehand, so everybody can prepare
	Start on Time and also end on time (I have work right afterwards)
Working norms	communicate openly and respectfully with each other espcially in case of conflicts or diverging opinions
	Ask questions as soon as you get stuck and don't try to grind through on your own
Coordination norms	Team Meeting: PO or SM
	Assignment: PO for the respective component
Communication norms	acknowledge within a day to direct questions (excluding maybe weekends), communicate via slack public except for e.g. meeting for pair programming
	As few tools as possible, . Its best to report issues as early as possible and be direct.
Consideration norms	Keep side conversations for the end of the meeting, so people who have to leave on time can do so
	We take everybody's opinion into consideration but final decision should be taken by voting
Cont. improvement norms	track progress: github board, evaluate work: code reviews etc?
	Suggestions for improvements are always welcome
Rewards	By receiving good feedback from the teaching team or achieving a milestone we can meet for a coffee! :)
Sanctions	conflicts / not meeting obligations probably not due to ill intent, so talk with the preson to find out reason and find a solution
	ideally everybody takes responsibility but this should be decided case by case.
Signatures	
Scrum Master	Rahil Soni
Product owner	Nahrain Gtari
Product owner	Andreas Freund
Software developer	Sebastian Alberto Rodriguez Schmidt
Software developer	Wiktor Pilarczyk
Software developer	Leon Wilms
Software developer	Michael Bauer
Software developer	Dorothea Ehrl
Software developer	Paul Roy

Product Vision	Project Mission
The reason of existence of the envisioned product (beyond this project). The product vision consists mainly in providing a reliable Flash Boot Loader that allows updating embedded software in automotive systems for users regardless of their technical expertise.	The mission of this particular project (in the context of the product vision). Our project mission consists mainly in developing a robust Flash Boot Loader with a user-friendly GUI that enables users, regardless of their technical expertise, to update the firmware on Aurix-based TC375 automotive embedded devices via the CAN Bus in an efficient and easy way. The project structure should allow Hella to adapt it to different platforms.

Term	Definition
Bootloader (BL)	Software, which is started before other Software to enable (remote) software updates of a device
Application Software (ASW)	Firmware for MCU
Electronic Control Unit (ECU)	

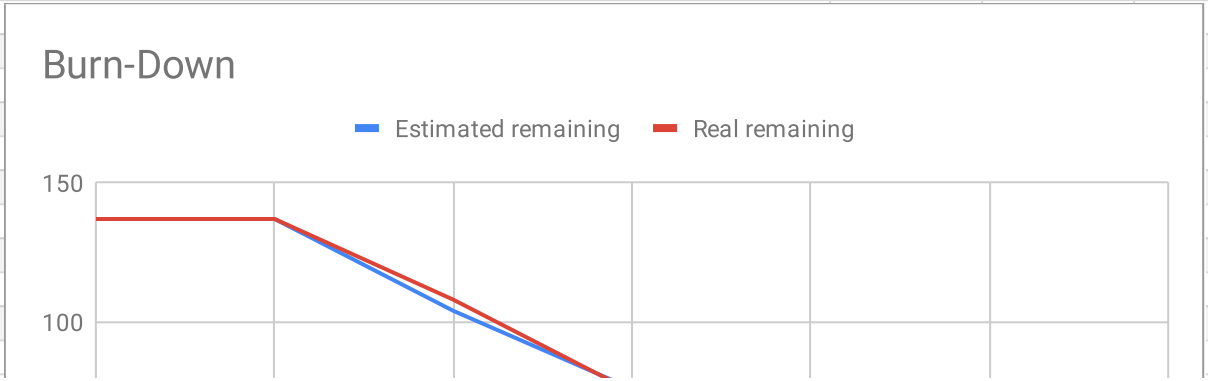
Sprint #	Sprint goal
1	None
2	None
3	None
4	Optional
5	Demonstrate end to end communication between GUI and MCU using some actual features
6	Get ready for mid-project release: UI cleanup & small refactoring
7	
8	
9	
10	
11	
12	
13	
14	
15	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			168	168		
Sprints						
7		Prepare for Flashing of the ECU	31	168	0	168
8		Flashing of the ECU	29	137	0	168
9		Can FD & Documentation	28	108	0	168
10		Testing (Reliability & Spec-Compliance)	29	80	0	168
11		Industry Partner Handover Tasks	25	51	0	168
12		Prepare for Final Release	26	26	0	168
				0		168
Features						
7		[WIN] Implement the CAN bus bitrate selection	3			
7		[MCU] Implement flash date display and flash persistence module	5			
7		[WIN] Add connectivity indicator	5			
7		[MCU] Adjust linker-script for bootloader	5			
7		[MCU] Create dummy ASW for demonstration purposes	5			
7		[WIN] Validation of firmware file	8			
8		Flash ECU Firmware	8			
8		Display Version Information before Flashing	5			
8		Check license of 2 .dll files in release .zip file	3			
8		Jump to ASW from Bootloader	5			
8		Reboot to Bootloader from ASW	8			
9		[WIN] Extend IsoTP for CAN FD	5			
9		[MCU] Extend IsoTP for CAN FD	5			
9		[MCU] CAN FD Driver	5			
9		[WIN] CAN FD Wrapper	5			
9		[WIN] Integrate CAN FD with the GUI	5			
9		Verify Firmware Update (Post Flashing)	3			
10		Create CANoe config for IsoTP, UDS compliance testing	8			

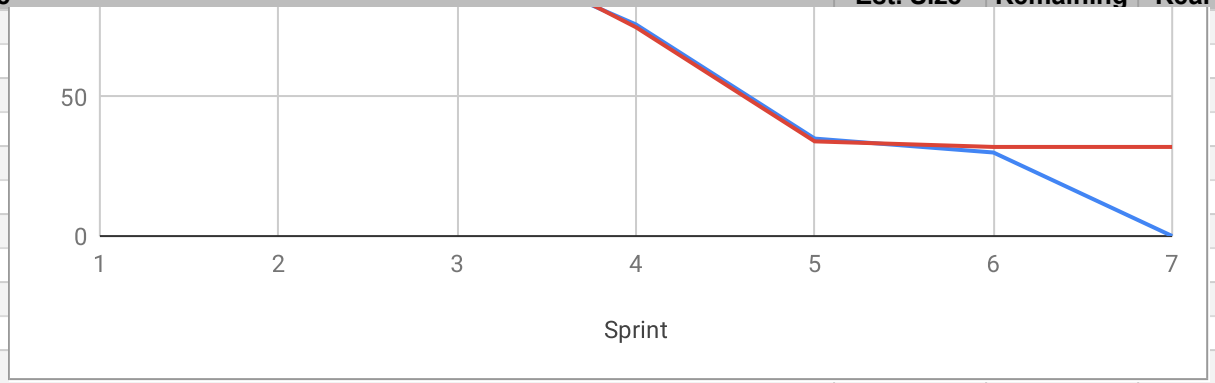
Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
10		Update User documentation	5			
10		Fix Errors found in CANoe	5			
10		Measure Boot Time Impact	3			
10		Reliability Testing (Power Failure during flashing, CAN Bus Failure)	8			
11		Create Porting Guide	5			
11		*Placeholder for Tasks from Industry Partner after 1st Handover Meeting in Sprint 9*	*20*			
12		Integrate User Documentation into GUI	3			
12		Add Setup Instructions into .zip Package	3			
12		*Placeholder for Tasks from Industry Partner after 2nd Handover Meeting in Sprint 11*	*20*			

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			137	137		
Sprints						
1		Research	0	137	0	137
2		Setup projects, High risk tasks	33	137	29	137
3		Combine modules	28	104	33	108
4		Get IsoTP & UDS working	41	76	41	75
5		Demonstrate end to end communication between GUI and MCU using some actual features	5	35	2	34
6		Get ready for mid-project release: UI cleanup & small refactoring	30	30	0	32
7				0		32
Features						
1	4	Align on a GUI framework				
1	1	Create Logo				
1	3	Get familiar with CAN-Bus, Aurix IDE, MCU programming				
2	13	Initialize software bill of materials	1		1	
2	10	[MCU] CAN driver	5		5	
2	9	[MCU] LED driver	5		2	
2	12	[Win] CAN Wrapper	5		5	
2	15	Initialize Readme file	1		1	
2	14	Create the description of the planned software architecture	3		2	
2	6	Create *.s19 to Binary converter	3		3	
2	5	Implement a function for file browsing, selecting, and simple processing	5		5	
2	11	[MCU] Flash driver	5		5	
3	34	[MCU] Build Process	3		2	
3	35	Implement UDS Communication Specification	3		3	
3	17	Implement a function to select CAN-HW	5		3	
3	19	Integrate CAN Wrapper with GUI	3		8	
3	21	Document / obtain documentation for relevant part of UDS	3		5	
3	33	[GUI] QT App Build process	3		5	
3	7	[UI] Implement "Updating in progress" indicator	2		2	
3	27	Design a Mockup for ECU Listing UI	3		3	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
3	25	[MCU] Create / modify existing project to demonstrate module functionality	3		2	
4	40	[MCU] Communication Layer (IsoTP)	8		8	
4	39	[WIN] Communication Layer (IsoTP)	5		8	
4	44	[MCU] UDS Tx	5		5	
4	43	[MCU] UDS messageInterpreter (Rx)	5		5	
4	31	[MCU] Implement Real Bus in CAN Driver	5		8	
4	42	[WIN] UDS Tx	5		3	
4	41	[WIN] UDS messageInterpreter (Rx)	5		3	
4	30	Review transitive include of e.g. "lib/vxlapi.h" in CAN_Wrapper	3		1	
5	18	Restart the ECU (soft reset) over GUI	3			
5	47	Build Process Video				
5	23	Packaging of Windows Aplication - .zip / Installer	2		2	
6	67	Clarify License Text	3			
6	66	[GUI] Add Clear Log Button	2			
6	75	[Bugfix] Investigate bytes lost maybe in IsoTP	5			
6	52	[Win] Fix CAN-Bus not opening, when already in use by another app, e.g. CANoe	1			
6	68	GUI changes	5			
6	51	Populate MCU List from CAN Bus	2			
6	73	Move buildscripts into subdirectories	1			
6	81	Midterm Refactoring and Bugfixing	3			
6	72	Update Software Architecture Diagram	3			
6	69	Refactoring of Drivers	5			



Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining



#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	Code review has been completed	MCU & GUI build and run	Manual Integration test passes for completed features
	Code has been merged	Release notes have been written	Architecture Diagram is up to date
	Feature branch has been deleted	Check licences again	Readme is up to date
	Update method signatures in architecture diagram	Check SBOM	Verify IDE / Env setup instructions
	MCU & GUI build after merge		

Type	Link / reference

#	Context	Name	Version	License	Comment
HW-1	Dev Kit	Aurix lite kit with TC375	V2		Aurix HW kit
HW-2	CAN Interface 1	Vector VN1611	/		HW for communication with CAN / CAN-FD
HW-3	CAN Interface 2	Vector CANcaseXL	/		HW for communication with CAN
SW-1	Windows GUI development	QT with MSVC2019 64 and cmake.	6.8.0	GPLv3	Software for developing the GUI
SW-2	MCU Aurix development and build	Aurix Development Studio (Aurix IDE)	v1.9.20	Boost Software License Version 1.0	Tasking Compiler can only be used in Aurix IDE
SW-3	Build GUI application	cmake	/		Tool to build the GUI application
SW-4	Build GUI application	MSVC22	/		Compiler to build the GUI application
SW-5	Build GUI application	Microsoft Windows SDK	/		Software development kits
SW-6	CAN Wrapper (Initial issue #12)	Vector XL-Driver Library	20.30.14	Free Library / Copyright Vector Informatik GmbH	Library includes vxlapi.h and DLL + LIB, Necessary to communicate with Vector CAN Adapter - Info about Lib see: https://www.vector.com/de/de/produkte/produkte-a-z/treiber-bibliotheken/xl-driver-library/#

Last Name	First Name	Value		#DIV/0!	#DIV/0!		
Wilms	**Leon**						
Rodriguez Schmidt	**Sebastian Alberto**						
Roy	**Paul**						
Pilarczyk	**Wiktor**						
Ehrl	**Dorothea**			0	No size		
Bauer	**Michael**			1	Trivial size		
				2	Small size		
Soni	**Rahil**			3	Medium size		
Gtari	**Nahrain**			5	Large size		
Freund	**Andreas**			8	Very large size		
				13	Too large (size)		
How to play planning poker							
1. Everyone type their number into their value field, don't hit return yet							
2. Someone, perhaps a product owner, count down 3.. 2.. 1..							
3. Then, everyone hit return to submit their value							