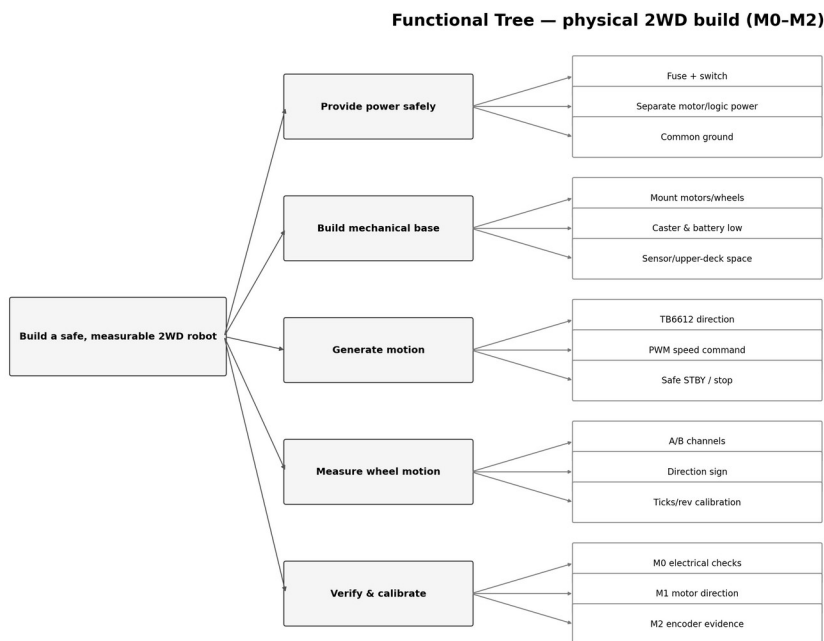


Fiziskā 2WD mācību robota uzbūves rokasgrāmata

Kursa “Programmēšanas un datorizētu sistēmu pamati” praktiskā platforma

References konfigurācija v11.0 — M0–M60 | 2026-08-21



Dokumenta statuss: Mācību un prototipēšanas materiāls. Šis nav safety-certified robot controller vai sertificētas elektroiekārtas projekts. Pirmās kustības pārbaudes veic ar paceltiem riteņiem un fiziski pieejamu barošanas slēdzi.

1. Mērķis un projekta loma kursā

Šīs rokasgrāmatas mērķis ir pārvērst iepriekš izveidoto 2WD programmatūras etalonprojektu par vienu konkrētu, atkārtojamu fizisko platformu, kuru var izmantot no pirmajām mehatronikas laboratorijām līdz ROS 2, SLAM, Nav2 un Machine Vision posmiem.

- Studentam jāredz pilna ķēde: prasība → Functional Tree → elektriskā/mehāniskā realizācija → firmware → mērījums → kalibrācija → ROS 2 → autonomija.
- M0–M2 apzināti neizmanto PID, SLAM vai AI. Vispirms jāpierāda barošana, kustība un mērījums.
- Visi vēlākie algoritmi balstās uz uzticamu M0–M2 aparatūras pamatu.

1.1. Sasniedzamais stāvoklis pēc M2

Kompetence	Pierādījums
Droša zemsprieguma barošana	Aizpildīts M0 mērījumu protokols; STBY safe-disable darbojas
Atvērta cikla motora vadība	Abi riteņi izpilda +/- komandas pareizajā fiziskajā virzienā
Enkoderu datu iegūšana	A/B kanāli rada stabilu signed tick skaitījumu
Kalibrēšana	ticks/rev noteikts ar 10 apgriezienu mērījumu, nevis tikai no datasheet
Dokumentēšana	Pinout, test case, Expected/Actual/PASS-FAIL un secinājums

2. References aparatūras konfigurācija

Lai studentu rezultāti būtu salīdzināmi, pamatlaboratorijām tiek definēts viens references komplekts. Alternatīvas drīkst izmantot, bet tad students dokumentē izmaiņas un atkārtoti nosaka kalibrācijas konstantus.

Poz.	Komponents	References detaļa	Daudzums	Kāpēc
1	Mikrokontroleris	Espressif ESP32-DEVKITC-32E	1	USB serial, daudz GPIO, 3.3 V loģika, Wi-Fi/BLE tālākam IoT
2	Motora draiveris	Pololu TB6612FNG carrier #713	1	Divi H-bridge kanāli, PWM, STBY, 1 A/channel continuous
3	Motori	Pololu #5138, 100:1 MP 6 V + 12 CPR encoder	2	Metāla reduktors, integrēts quadrature encoder, samērīga strāva
4	Riteņi	Pololu 60×8 mm pair #1423 vai tās pašas ģeometrijas variants	1 pair	3 mm D-shaft, nominal Ø60 mm
5	Motora kronšteini	Pololu standard micro metal gearmotor bracket pair #1086 / ekvivalents	1 pair	Atkārtojama motora mehāniskā fiksācija
6	Enkoderu kabeli	Pololu #4762, 6-pin JST-SH single-ended	2	Krāsu kodēti A/B/Vcc/GND + motora vadi
7	Balsta ritenis	Pololu #951 3/8" metal ball caster / ekvivalents	1	Mazs balsta punkts 2WD šasijai
8	Motoru barošana	4×AA NiMH holder + 4 matched cells	1	Zems spriegums un samazināts ātrums mācību laboratorijai
9	Aizsardzība	2 A fuse + holder, master switch	1+1	Fiziska enerģijas atslēgšana un vadu aizsardzība
10	IMU (M5)	MPU6050/GY-521	1	Esošais firmware atbalsts un labs mācību sensors
11	Regulators (vēlāk)	Pololu S7V8F5 #2123	1 optional	Untethered 5 V logic power from battery
12	LiDAR (M9+)	SLAMTEC RPLIDAR A1-class	1 optional	2D /scan, SLAM/Nav2 laboratorijas

Svarīga izvēle: TB6612FNG references carrier ir paredzēts 4.5–13.5 V motoru barošanai, 2.7–5.5 V loģikai un Pololu carrier norāda 1 A nepārtrauktu strāvu uz kanālu. References #5138 motora teorētiskā stall strāva ir 0.67 A pie 6 V. Stall tomēr nav darba režīms un laboratorijā to apzināti nerada.

2.1. Sākotnējās ģeometrijas konstantes

`WHEEL_RADIUS_M = 0.030 m`

`TRACK_WIDTH_M = 0.160 m`

`ENCODER_TICKS_PER_WHEEL_REV = 301.11 // only a starting value`

Šie nav “pareizie uz mūžu” skaitļi. Riteņa efektīvais rādiuss mainās ar riepas deformāciju/slodzi, track width efektīvā vērtība pagriezienos ietver slīdēšanu, bet enkodera ticks/rev ir atkarīgs arī no firmware skaitīšanas metodes. M2 un M4 tos pārvērš par mērītiem parametriem.

3. Kāpēc enkodera sākuma konstante ir 301.11?

Pololu #5138 apraksta 12 CPR incremental quadrature enkoderi uz motora vārpstas un 100.37:1 reduktoru. Pilnas quadrature skaitīšanas gadījumā 12 counts/rev tiek iegūti, skaitot abus kanālus un abas malas. Mūsu references ESP32 ISR pašlaik aktivizējas tikai uz A kanāla RISING malas un B kanālu lasa virziena noteikšanai.

full quadrature motor CPR = 12

A-rising-only counts per motor rev = $12 / 4 = 3$

gear ratio = 100.37

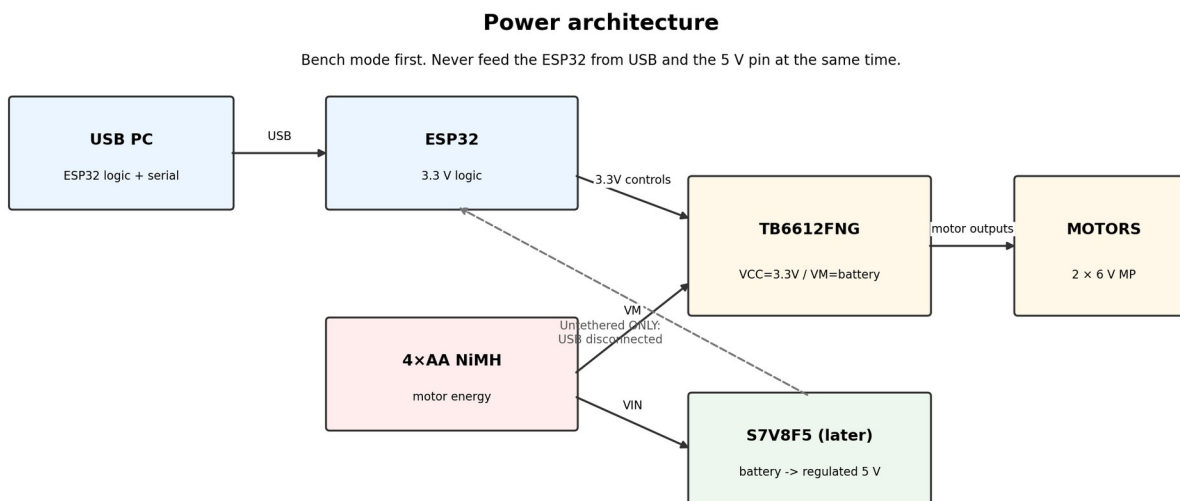
starting wheel ticks/rev = $3 \times 100.37 = 301.11$

Pasniedzējam: Šī ir ļoti laba vieta, kur izskaidrot, ka "CPR" bez konteksta var būt bīstams termins: dažādi ražotāji/bibliotēkas ar CPR/PPR/ticks var domāt dažādas edge-decoding konvencijas. Studentam jāprot sasaistīt datasheet ar konkrēto interrupt algoritmu.

3.1. M2 galvenā doma

301.11 ir tikai teorētiska sākuma konstante, kas ļauj pirmo reizi palaist sistēmu. Pēc tam students fiziski pagriež riteni 10 pilnus apgriezienus, atkārto mērījumu vismaz trīs reizes un ievada mērīto vērtību.

4. Elektrobarošanas arhitektūra



4.1. attēls. Bench un untethered barošanas principi.

4.1. Bench režīms — obligāts M0–M6

- ESP32 saņem barošanu un serial komunikāciju tikai no USB.
- 4x AA NiMH pack baro tikai TB6612 VM / motoru daļu.
- ESP32 GND, TB6612 GND un battery – ir savienoti vienā common GND.
- TB6612 VCC un enkoderu Vcc tiek baroti no ESP32 3V3.
- Master switch un 2 A fuse atrodas motoru baterijas + vadā.

Kritiska prasība: Espressif dokumentācija norāda ESP32-DevKitC barošanas opcijas USB, 5V/GND un 3V3/GND kā savstarpēji izslēdzošas. Tāpēc pamatlaboratorijās baterijas 5 V regulatoru vispār nepievieno ESP32 5V pinam, kamēr USB ir pieslēgts.

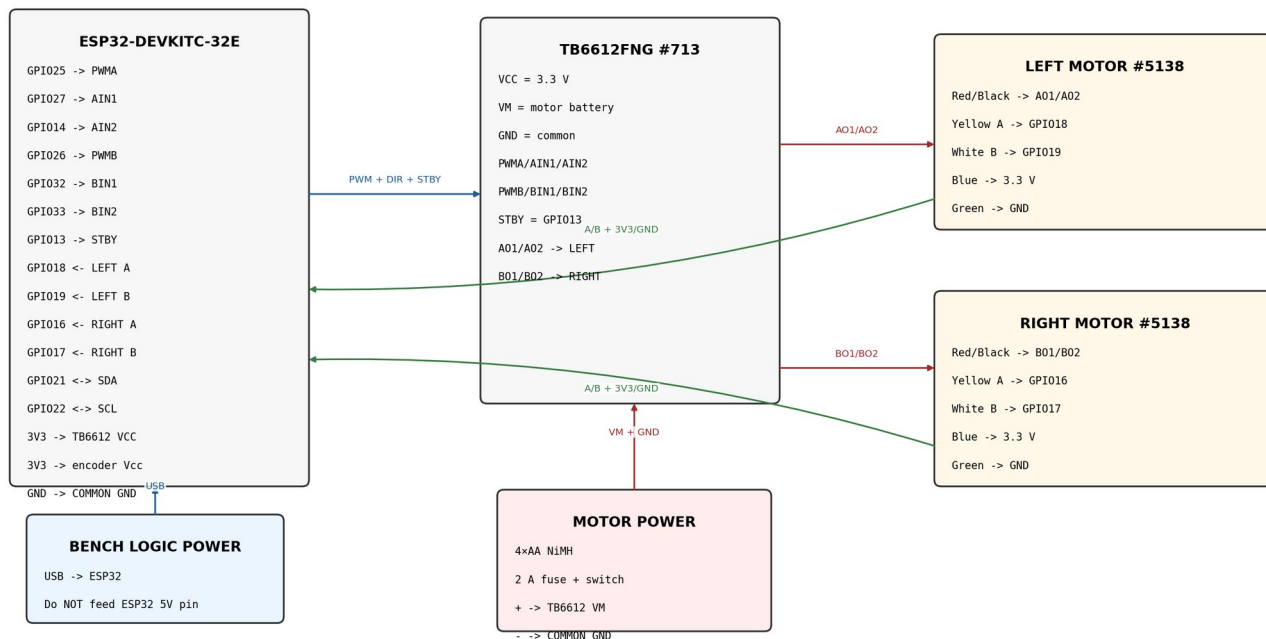
4.2. Untethered režīms — tikai pēc pamatlaboratorijām

S7V8F5 var izmantot, lai no 4-cell battery pack iegūtu stabilus 5 V. Pirms 5 V pievienošanas ESP32 5V pinam USB kabelis ir fiziski jāatvieno. Vēl drošākā mācību versijā regulatora 5 V vads uz ESP32 ir ar atsevišķu noņemamu jumperi.

5. Pilna elektroinstalācija M0-M2

2WD reference wiring — M0-M2 bench mode

Motor power and USB logic power are separate; all grounds are common.



5.1. attēls. References wiring M0–M2 bench režīmā.

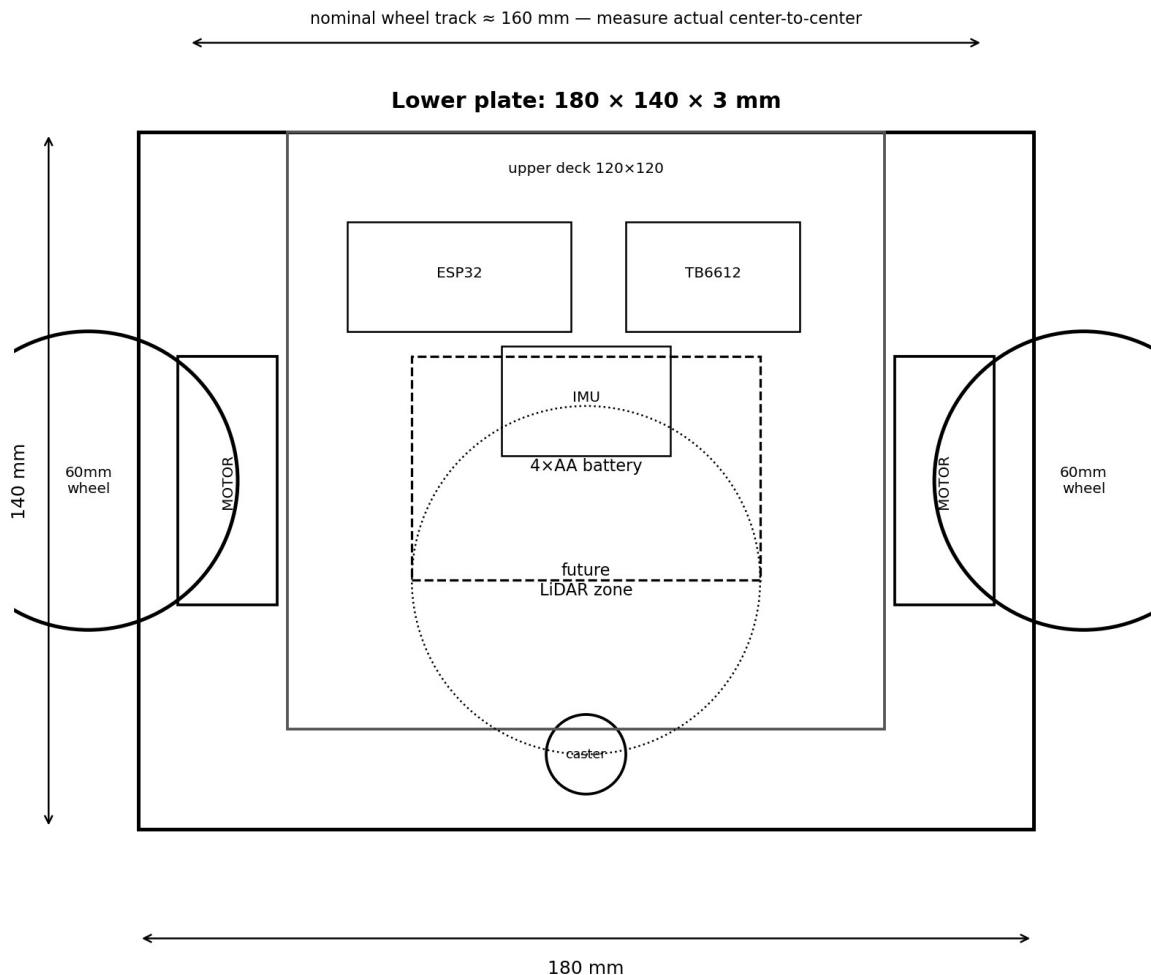
Funkcija	ESP32	TB6612 / sensors	Piezīme
Left PWM	GPIO25	PWMA	20 kHz baseline PWM
Left direction 1	GPIO27	AIN1	
Left direction 2	GPIO14	AIN2	
Right PWM	GPIO26	PWMB	20 kHz baseline PWM
Right direction 1	GPIO32	BIN1	
Right direction 2	GPIO33	BIN2	
Driver enable	GPIO13	STBY	HIGH = enabled; internal pull-down on carrier
Left encoder A	GPIO18	yellow A	RISING interrupt
Left encoder B	GPIO19	white B	direction sample
Right encoder A	GPIO16	yellow A	RISING interrupt
Right encoder B	GPIO17	white B	direction sample
Logic	3V3	TB6612 VCC	3.3 V
Encoder Vcc	3V3	blue	2.7–18 V supported by Pololu encoder; use 3.3 V
Common ground	GND	TB6612 GND + green + battery –	mandatory
Motor supply	—	TB6612 VM	4×AA pack through fuse + switch

5.1. Pololu #4762 kabeļa krāsu kods

Pin	Krāsa	Funkcija	Savienojums references robotā
1	Green	encoder GND	Common GND
2	White	encoder B	GPIO19 left / GPIO17 right
3	Yellow	encoder A	GPIO18 left / GPIO16 right
4	Blue	encoder Vcc	ESP32 3V3
5	Black	motor M2	TB6612 A/B output terminal
6	Red	motor M1	TB6612 A/B output terminal

6. Mehāniskā platforma

Reference mechanical layout — top view



6.1. attēls. Ieteicamais šasijas izvietojums; gala caurumi jāpārbauda pret reālām detaļām.

References zemākā plāksne ir $180 \times 140 \times 3$ mm. Tā ir pietiekami liela, lai saglabātu vietu baterijai, balsta ritenim un 120×120 mm augšējam klājam, bet pietiekami maza, lai 60 mm riteņi joprojām definētu kompaktu mācību robotu.

- Baterija — centrā un zemu, lai samazinātu masas centra augstumu.
- IMU — pēc iespējas stingri un horizontāli, tuvāk robota centra plaknei.
- ESP32/TB6612 — augšējā klājā ar pieejamu USB un master switch.
- LiDAR — vietu rezervē jau šasijas sākumā, bet sensoru montē tikai M9+.
- Motora kronšteini — OpenSCAD failā ir universāli sloti. Pirms gala urbšanas veic dry-fit ar konkrēto bracket.

Ražošanas fails: `mechanical/chassis\_reference.scad` ir parametriska sagatave, nevis sertificēts ražošanas rasējums. Motora kronšteinu un caster sloti apzināti ir pielāgojami.

7. Montāžas secība

1. Izgatavo/izgriez zemāko šasijas plāksni un attīri malas.
2. Piemēri motora kronšteinus un tikai pēc dry-fit fiksē gala caurumus/slotus.
3. Uztādi motorus un 60 mm riteņus; pārbaudi brīvu rotāciju.
4. Uztādi caster un panāc aptuveni horizontālu šasiju.
5. Uztādi bateriju turētāju zemākajā praktiskajā vietā.
6. Uztādi master switch un fuse tā, lai pasniedzējs var ātri atslēgt motoru enerģiju.
7. Uz augšējā klāja uztādi ESP32 un TB6612; saglabā USB pieejamību.
8. Izveido common GND sadales punktu.
9. Veic M0 ar motoriem sākotnēji atvienotiem no draivera.
10. Tikai pēc M0 PASS savieno motorus un pārej uz M1.
11. Tikai pēc M1 PASS pieslēdz/validē encoder signal wiring un pārej uz M2.

8. Pirmās palaišanas procedūra — “no nulles līdz pirmajai drošajai kustībai”

8.1. Pirms jebkādas barošanas

- Vizual inspection: nav pliku vadu, vaļīgu skrūvju vai polaritātes neskaidrību.
- Multimetrs continuity režīmā: battery + pret GND nedrīkst būt low-resistance short.
- STBY vadam jābūt uz GPIO13 un programmas default stāvoklim — motor disabled līdz initialization.
- Riteņi pacelti virs darba virsmas.

8.2. Loģika bez motora enerģijas

12. Master motor switch OFF.
13. Pieslēdz ESP32 USB.
14. Upload firmware.
15. Atver serial monitor un pārbaudi boot bez reset loop.
16. Pārbaudi GPIO/pinout pret `config.h`.

8.3. Motor driver bez pievienota motora

17. Pievieno motor battery un ieslēdz switch.
18. Izmēri VM pret GND.
19. Izmēri VCC pret GND (~3.3 V).
20. Pārbaudi, ka STBY LOW atstāj outputs safe.
21. Ja kaut kas silst bez motora slodzes — OFF un diagnosticē.

8.4. Pirmais motor test

22. Power OFF. Pieslēdz tikai LEFT motor.
23. Power ON, komanda apmēram 20–25% uz ≤2 s.
24. Pārbaudi virzienu un reset/brownout neesamību.
25. Atkārti RIGHT motoram.
26. Kad abi atsevišķi PASS, pieslēdz abus un veic M1.

9. M0 — Power, wiring and safe stop (90 min)

M0 nav “garlaicīga sagatavošana”. Tā ir pirmā īstā sistēminženierijas laboratorija: students pierāda, ka enerģija ir kontrolēta, references mezgli ir pareizi un sistēmu var droši atslēgt.

Laiks	Pasniedzējs	Students / pierādījums
0–10 min	Izskaidro logic power, motor power, common GND un īssavienojuma risku.	Atzīmē power domains shēmā.
10–25 min	Demonstrē multimetra continuity un DC voltage režīmus.	Veic pre-power checklist.
25–45 min	Vada motor battery → fuse → switch → VM wiring.	Izveido un partnerim pārbauda ķēdi.
45–60 min	Vada USB logic + common GND + TB6612 VCC/STBY.	Izmēra VCC, VM, dokumentē Actual.
60–75 min	Demonstrē STBY safe-disable.	Izpilda STBY test case.
75–90 min	Review + kļūdu diskusija.	Secinājums: kāda atteice būtu bīstamākā?

9.1. Ko stāstīt

Datorsistēmā GND nav “mistisks mīnuss”; tas ir references potenciāls, pret kuru tiek interpretēti loģiskie signāli. Ja ESP32 un TB6612 nav kopīga reference, HIGH/LOW interpretācija var kļūt neprognozējama pat tad, ja abas ierīces atsevišķi ir ieslēgtas.

STBY ir programmatūras kontrolēts draivera enable mehānisms. Tas ir vērtīgs mācību drošības slānis, bet tas neaizstāj fizisko battery switch: bojāts firmware, īssavienojums vai nepareizs wiring var atrasties ārpus programmas kontroles.

9.2. M0 testa protokols

Test case	Expected	Actual	PASS/FAIL
Battery + to GND before power	No short / high resistance	_____	_____
ESP32 boot via USB	Stable boot, serial output	_____	_____
TB6612 VCC	~3.3 V	_____ V	_____
TB6612 VM	Battery voltage	_____ V	_____
STBY LOW	Driver disabled	_____	_____
Thermal check, no motor load	No abnormal heating	_____	_____

10. M1 — Open-loop motor control (90 min)

M1 mērķis nav “robotam pabraukt”. Mērķis ir izveidot skaidru un reproducējamu sakarību starp software command un fizisko aktuatora virzienu/ātrumu.

Laiks	Saturs	Pierādījums
0–15 min	H-bridge un direction truth-table ideja	Students spēj paredzēt polarity effect
15–30 min	PWM kā duty-cycle kontrole	Paredz ātruma tendenci 25/50/75%
30–50 min	LEFT motor test	Direction + 3 PWM punkti
50–70 min	RIGHT motor test	Direction + 3 PWM punkti
70–80 min	Zero + STBY	Stop behavior dokumentēts
80–90 min	Review/debug	Motor sign settings fiksēti

10.1. Tipiskās kļūdas

- Viens motors griežas pretēji — maina tikai vienu definētu virziena konvenciju, nevis vienlaikus pārliet vadus un sign flags.
- ESP32 restartējas pie motora starta — meklē barošanas kritumu, common GND problēmu, motora trokšņus vai nepareizu power topology.
- Motor negriežas — pārbauda STBY, VM, direction input un PWM atsevišķi, nevis “maina visu”.

10.2. M1 datu tabula

Side	Command	Expected direction	Observed	Noise/reset?	PASS
Left	+25%	Forward wheel rotation	_____	_____	___
Left	+50%	Forward, faster	_____	_____	___
Left	+75%	Forward, faster	_____	_____	___
Right	+25%	Forward wheel rotation	_____	_____	___
Right	+50%	Forward, faster	_____	_____	___
Right	+75%	Forward, faster	_____	_____	___

11. M2 — Encoder acquisition and calibration (90 min)

M2 ir pāreja no “vadām” uz “mērām”. No šī brīža robotam parādās state feedback pamats. Vēl nav closed loop, bet ir uzticams sensors, kuru M3 var izmantot PID.

Laiks	Saturs	Pierādījums
0–15 min	Quadrature, A/B fāze, direction	Student izskaidro why two channels
15–30 min	Firmware ISR un edge counting	Sasaista 12 CPR ar 301.11 start value
30–45 min	Manual forward/backward check	Signed ticks correct
45–65 min	10-rev test ×3, left	ticks/rev dataset
65–80 min	10-rev test ×3, right	ticks/rev dataset
80–90 min	Aprēķins + config update	Measured constant + conclusion

11.1. Aprēķina veidne

$\text{ticks_per_rev_i} = \text{abs}(\text{ticks_after} - \text{ticks_before}) / 10$

$\text{left_final} = \text{median}(\text{left_trial_1}, \text{left_trial_2}, \text{left_trial_3})$

$\text{right_final} = \text{median}(\text{right_trial_1}, \text{right_trial_2}, \text{right_trial_3})$

Ja firmware izmanto vienu kopēju `ENCODER\_TICKS\_PER\_WHEEL\_REV`, izvēlas pamatotu kopējo vērtību (piem., abu pušu vidējo/mediānu) un dokumentē atlikušās atšķirības. Ja vēlāk firmware tiek paplašināts ar atsevišķām left/right konstantēm, var izmantot katras puses rezultātu atsevišķi.

Side	Trial 1 / 10 rev	Trial 2	Trial 3	ticks/rev final
Left				
Right				

12. M0-M2 acceptance gate

Studentu grupa drīkst pāriet uz M3 wheel-speed PID tikai tad, ja zemākā tabula ir aizpildīta ar pierādījumiem.

ID	Prasība	Pierādījums	Statuss
A-01	Nav short pirms power-on	Continuity measurement	PASS / FAIL
A-02	Motor power fiziski atslēdzams	Master switch test	PASS / FAIL
A-03	STBY disable reproducējams	Test case	PASS / FAIL
A-04	Left software + = forward wheel	M1 table	PASS / FAIL
A-05	Right software + = forward wheel	M1 table	PASS / FAIL
A-06	No brownout at normal open-loop test	Serial observation	PASS / FAIL
A-07	Left forward ticks positive	M2 evidence	PASS / FAIL
A-08	Right forward ticks positive	M2 evidence	PASS / FAIL
A-09	ticks/rev measured ×3 per side	M2 table	PASS / FAIL
A-10	config.h updated from evidence	Git diff / code review	PASS / FAIL

13. Troubleshooting matrica

Simptoms	Iespējamais cēlonis	Pārbaude	Korekcija
ESP32 reset, kad motors sākas	Power dip/noise/common GND	Serial brownout + measure supply	Atdala power domains, pārbauda GND, wiring, decoupling
Motor negriežas	STBY LOW / no VM / wrong direction pins	Measure STBY, VM, PWM	Labot vienu kļūdu vienlaikus
Motor griežas pretēji	Polarity/sign convention	M1 single-wheel test	Swap motor terminal pair VAI motor sign, dokumentē
Ticks = 0	A/B/Vcc/GND kļūda	Manual rotation + GPIO monitor	Pinout/cable correction
Ticks only one direction / erratic	A/B wrong or floating	Slow hand rotation	Fix A/B/GND; shorter clean routing
Forward ticks negative	Encoder sign convention	Hand rotate forward	Change encoder sign only
301.11 neatbilst mērījumam	Real decoder/gear/wheel variation	10-rev M2 test	Use measured value
Robot later curves with equal wheel command	Motor asymmetry / wheel radius / friction	Log wheel speed independently	M3 PID + M4 geometry calibration

14. Pasniedzēja sagatavošanas checklist

- Vismaz vienu references robotu uzbūvēt un M0–M2 iziet pirms studentu laboratorijas.
- Sagatavot rezerves fuse, jumper wires, motor cable un vienu rezerves TB6612/ESP32.
- Katram darba galdam: multimetrs, izolēts zemsprieguma battery holder un fiziski pieejams switch.
- Uz šasijas marķēt LEFT/RIGHT, FRONT un battery polarity.
- Firmware repo freeze uz konkrētu commit pirms laboratorijas.
- Sagatavot Expected/Actual/PASS-FAIL lapas vai digitālo veidni.
- Neļaut studentiem vienlaikus mainīt wiring un software sign, ja tiek meklēta viena direction kļūda.
- M2 kalibrācijas rezultātu glabāt repo kopā ar robot ID, datumu un izmantoto firmware commit.

14.1. Ieteicamais robotu identifikators

Robot ID: RBT-01

Hardware revision: HW-A

Firmware commit: <git hash>

M2 encoder calibration date: YYYY-MM-DD

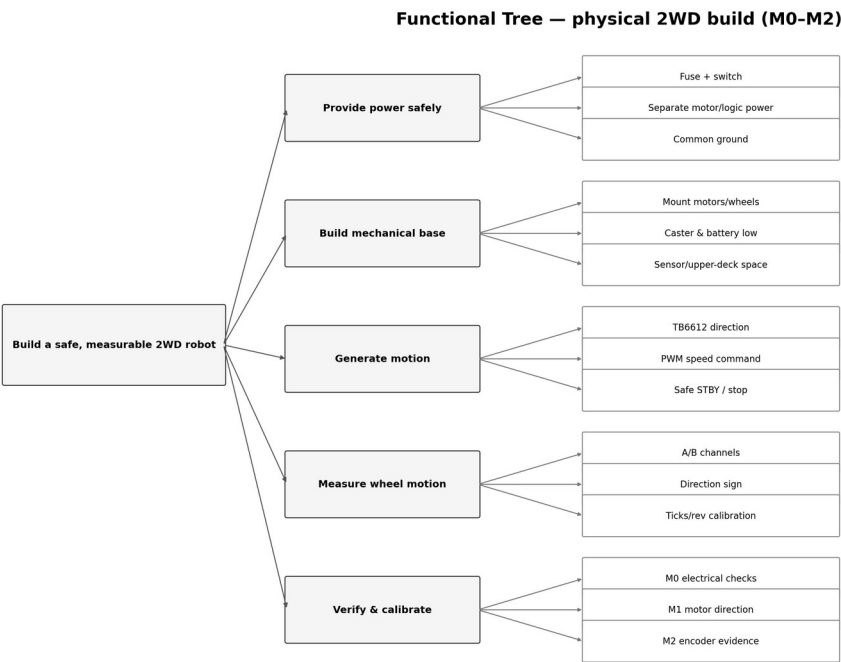
Left ticks/rev: ...

Right ticks/rev: ...

Wheel effective radius (M4): ...

Track width effective (M4): ...

15. Kā M0–M2 turpinās līdz autonomam robotam



15.1. attēls. Fiziskās platformas funkcionālā dekompozīcija.

Milestone	Kas tiek pievienots	Kāpēc M0–M2 ir priekšnoteikums
M3	Wheel-speed PID	PID nevar strādāt bez signed un calibrated wheel feedback
M4	Odometrija	Distance/yaw integrācija paļaujas uz ticks/rev un wheel geometry
M5	IMU	Sensor fusion salīdzina IMU ar wheel state
M7	ROS 2 bridge	ROS saņem raw data no pierādītas aparatūras
M8	EKF	Covariance tuning nav jēgas, ja raw sensor convention ir nepareizs
M9–M10	LiDAR + SLAM	Map kļūdas bieži sākas no sliktas odometrijas/TF
M11	Nav2	Planner/controller pieņem, ka robot base kustības modelis ir validēts
M12	Machine Vision	AI perception tiek pieslēgts sistēmai ar jau stabilu motion stack

16. Tehnisko avotu kopsavilkums

Avoti pārbaudīti 2026-08-19. Cenas un noliktavas statuss ir mainīgi; pirms pasūtījuma tos pārbauda no jauna.

Espressif ESP32-DevKitC V4 user guide:

https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user_guide.html

Pololu #5138 motor + encoder: <https://www.pololu.com/product/5138>

Pololu TB6612 #713: <https://www.pololu.com/product/713>

Pololu 60×8 wheel #1423: <https://www.pololu.com/product/1423>

Pololu bracket #1086: <https://www.pololu.com/product/1086>

Pololu caster #951: <https://www.pololu.com/product/951>

Pololu encoder cable #4762: <https://www.pololu.com/product/4762>

Pololu S7V8F5 #2123: <https://www.pololu.com/product/2123>

SLAMTEC RPLIDAR A1 spec: <https://www.slamtec.com/en/lidar/a1spec>

Gazebo Jetty ROS guidance: https://gazebo-sim.org/docs/jetty/ros_installation/

16.1. References cenu snapshot (nevis iepirkuma garantija)

Komponents	Snapshot	Valūta / piezīme
ESP32-DEVKITC-32E	~8.44	EUR, DigiKey Latvia result snapshot
Pololu #5138	29.95	USD direct Pololu, qty 1
Pololu TB6612 #713	4.95	USD direct Pololu, qty 1
Pololu 60×8 pair #1423	6.75	USD direct Pololu, qty 1
Pololu standard bracket pair #1086	2.95	USD direct Pololu, qty 1
Pololu caster #951	2.95	USD direct Pololu, qty 1
Pololu cable #4762	2.10 each	USD direct Pololu, qty 1
Pololu S7V8F5 #2123	9.95	USD direct Pololu, optional

17. M3–M6 — vadības, odoetrijas, IMU un failsafe bloks

M0–M2 pierādīja enerģiju, kustību un enkodera mērījumu. M3–M6 pievieno slēgtu vadības cilpu, pozīcijas novērtējumu, inerciālo mērījumu interpretāciju un drošu komunikācijas uzvedību. Katrs posms ir quality gate: nākamo posmu sāk tikai pēc izmērāma PASS pierādījuma.

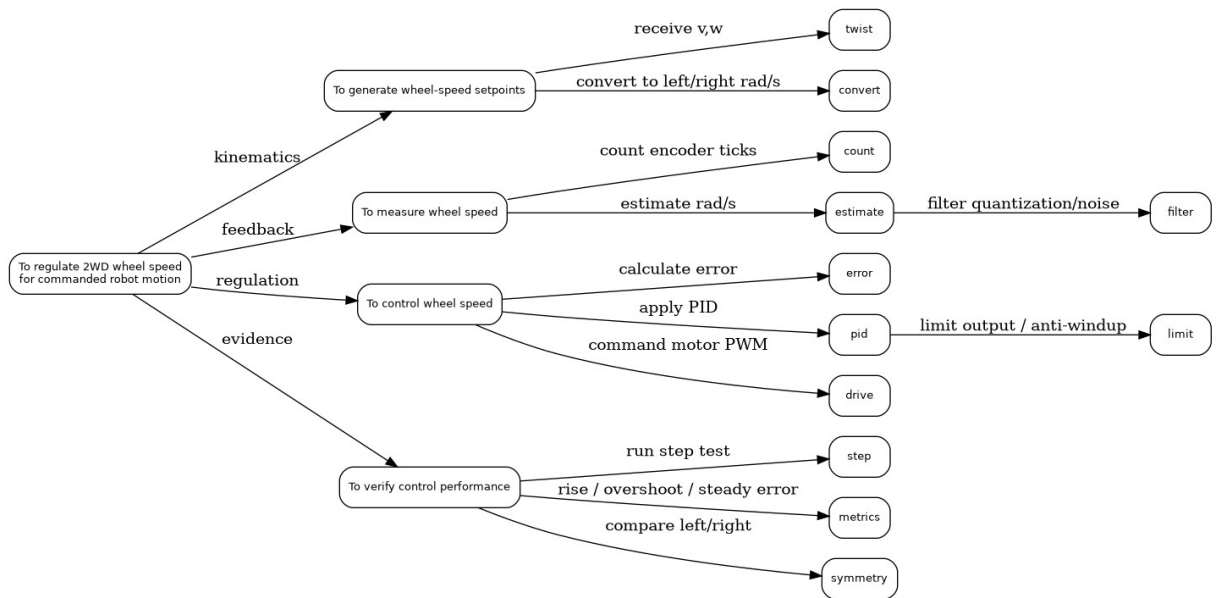
Command -> kinematics -> wheel PID -> motor -> encoder -> speed feedback

encoder ticks -> differential-drive odometry -> x, y, theta

MPU6050 -> calibrated angular rate -> relative yaw comparison

host commands -> parser -> command-age monitor -> failsafe stop

18. M3 — Wheel-speed PID un taisnas kustības regulēšana



18.1. attēls. M3 Functional Tree — wheel-speed closed-loop control.

18.1. Teorija, ko stāstīt

Open-loop PWM nenosaka riteņa ātrumu. Tas nosaka tikai vadības signālu. Faktisko ātrumu ietekmē baterijas spriegums, berze, reduktors, slodze un mehāniskā asimetrija. Closed-loop sistēma mēra rezultātu un labo kļūdu.

$$\text{error} = \text{setpoint} - \text{measured_speed}$$

$$u = K_p \cdot \text{error} + K_i \cdot \text{integral}(\text{error} \cdot dt) + K_d \cdot d(\text{error})/dt$$

v4.0 firmware PID papildināts ar conditional integration anti-windup. Tas neļauj integrāļa stāvoklim nekontrolēti augt, ja izeja jau ir piesātināta un kļūda turpina virzīt regulatoru tajā pašā virzienā.

Enkodera ātruma mērījums ir kvantēts. Pie neliela ātruma īsā 10 ms logā var būt tikai daži tick, tāpēc firmware izmanto vienkāršu first-order low-pass filtru. Tas samazina trokšņainumu, bet pievieno aizturi — svarīgs trade-off.

18.2. Praktiskā secība

1. Pirmo PID testu veic ar droši paceltiem riteņiem vai uz roller fixture.
2. Ar ? saglabā konfigurācijas snapshot.
3. Sāc ar P-only regulatoru un nelielu lineāro ātrumu.
4. Izmanto pid_step_test.py un saglabā C telemetry logu.
5. Aprēķini rise time, overshoot, steady-state error un IAE ar pid_metrics.py.
6. Pievieno Ki tikai pēc P-only datu analīzes.
7. Salīdzini LEFT/RIGHT steady-state ātrumu un controller output.
8. Tikai pēc stabila bench rezultāta veic 2–3 m taisnas kustības testu uz grīdas.

```
python host_tools/pid_step_test.py --port /dev/ttyUSB0 --speed 0.18 --output results/M03.csv
```

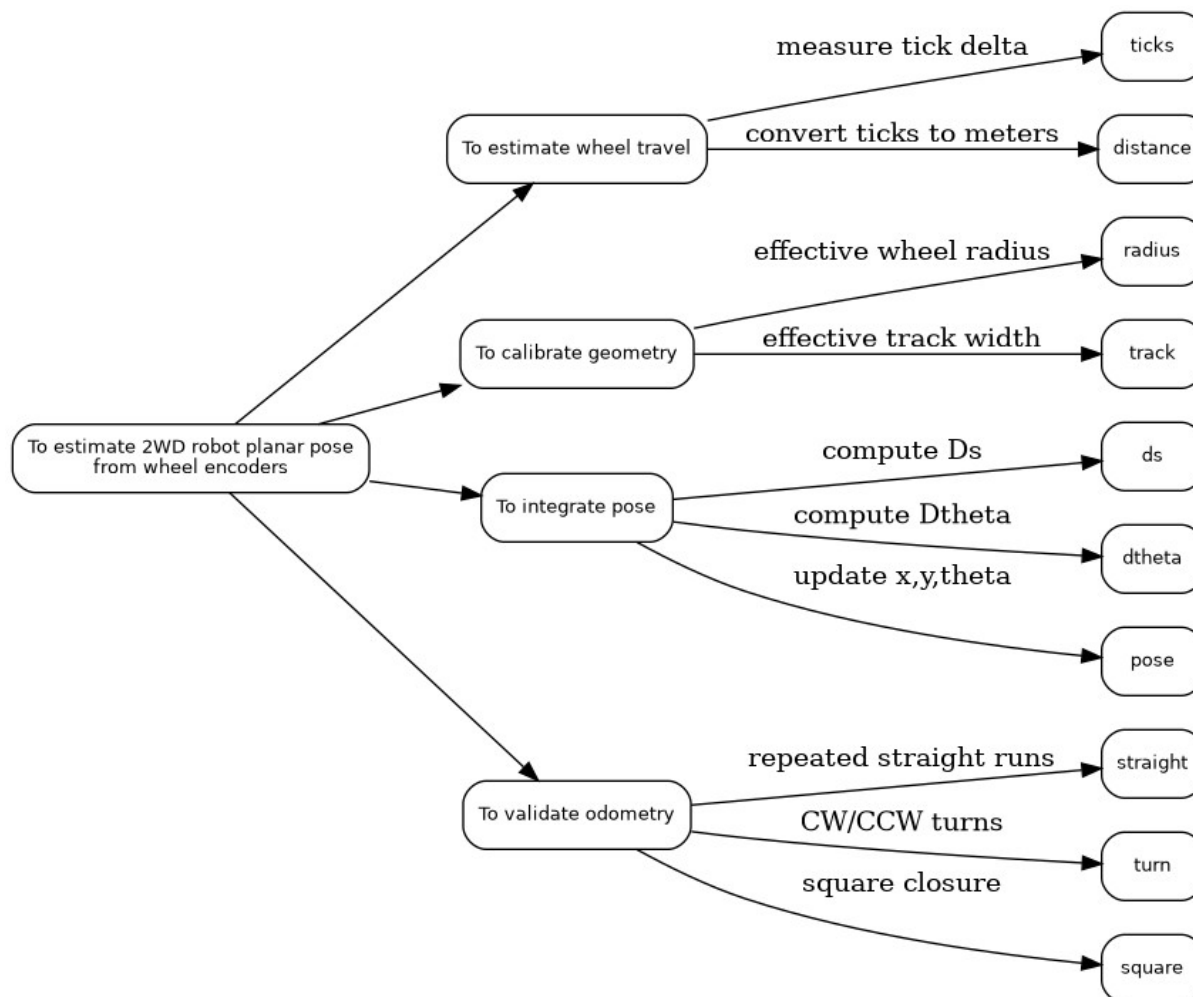
```
python host_tools/pid_metrics.py results/M03.csv
```

Pierādījums	Expected	Actual	PASS/FAIL
P-only logs	CSV + metrics		
PI/PID logs	CSV + metrics		
No sustained oscillation	YES		
Saturation understood	YES		
Left/right asymmetry reported	Numerical		
Gain choice justified	YES		

18.3. Pasniedzēja akcenti

- Neļaut studentam “tūnēt ar aci” bez saglabātiem logiem.
- Ja viens motors ir būtiski lēnāks, vispirms meklēt mehāniku, barošanu un enkodera kļūdas, nevis slēpt problēmu ar milzīgu K_i .
- Paskaidrot, ka filtrēts signāls nav “patiesāks” par nefiltrētu; tas ir kompromiss starp noise un delay.
- Uzsvērt, ka PID parametri pieder konkrētai sistēmas konfigurācijai — masa, riteņi, barošana un mehānika ir daļa no plant.

19. M4 — Differential-drive odometrija un ģeometrijas kalibrācija



19.1. attēls. M4 Functional Tree — wheel odometry and geometry calibration.

19.1. Teorija, ko stāstīt

$$\text{meters_per_tick} = 2\pi r / \text{ticks_per_rev}$$

$$Dl = \text{delta_left_ticks} * \text{meters_per_tick}$$

$$Dr = \text{delta_right_ticks} * \text{meters_per_tick}$$

$$Ds = (Dr + Dl)/2$$

$$Dtheta = (Dr - Dl)/b$$

Wheel radius r un track width b ir effective model parameters. Katalogā vai CAD izmērīta vērtība ir sākumpunkts, nevis automātiski labākais odometrijas parametrs. Riepas deformācija, grīda un montāža maina efektīvo ģeometriju.

19.2. Kalibrācijas secība

1. Veic vismaz piecus garus taisnus braucienus un salīdzini physical distance ar odometry distance.
2. Aprēķini wheel radius correction: $r_new = r_old * \text{physical} / \text{odometry}$.
3. Veic vairākus CW un CCW in-place turn testus.

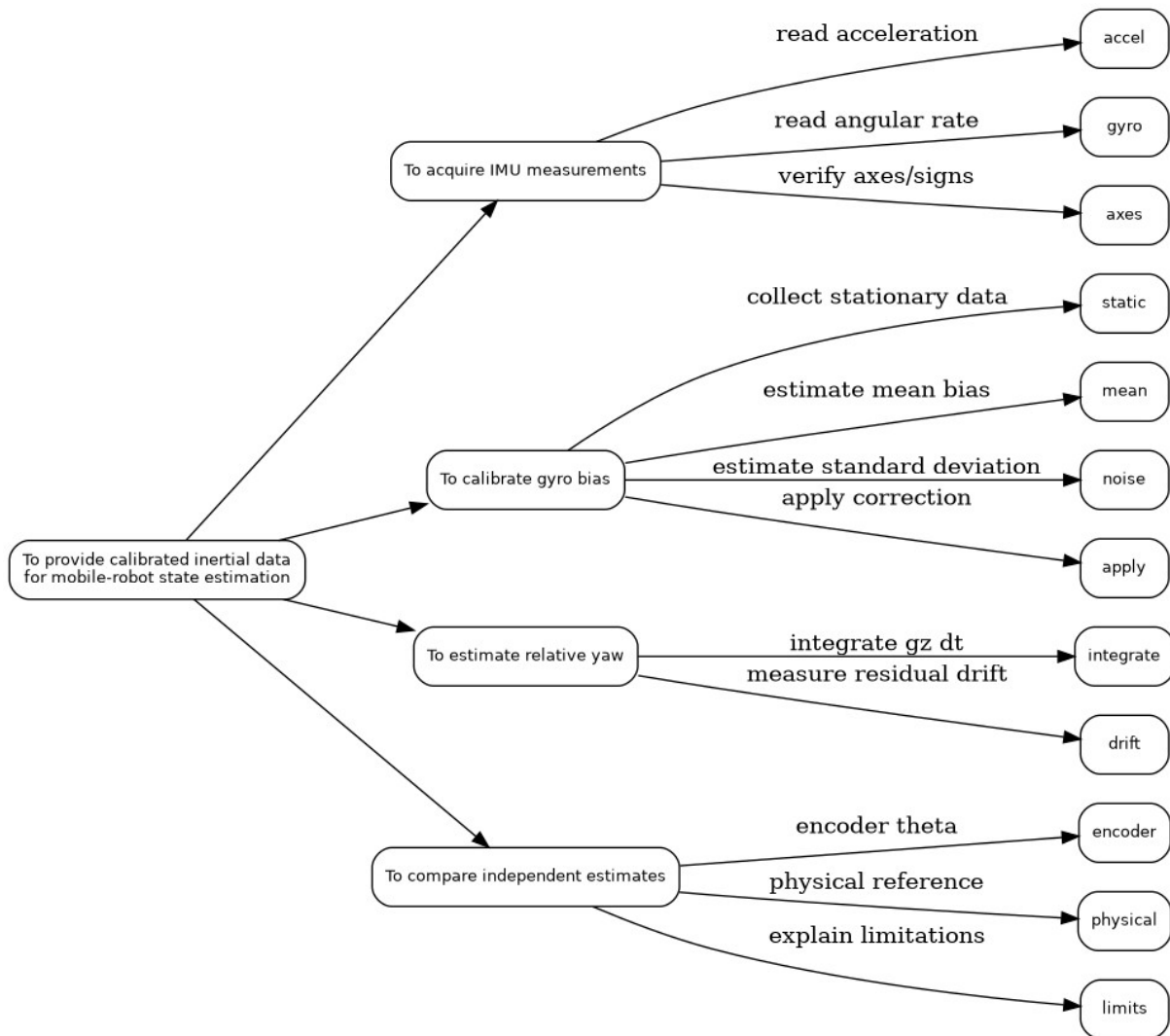
4. Aprēķini track width correction: $b_{\text{new}} = b_{\text{old}} * \text{odometry_angle} / \text{physical_angle}$.
5. Pēc parametru labojuma izpildi square-path closure testu.
6. Atsevišķi ziņo accuracy un repeatability; neizvēlies tikai labāko trial.

Test	Trials	Galvenais rezultāts	PASS/FAIL
Straight distance	≥ 5	radius correction + spread	
CW turn	≥ 3	track-width correction	
CCW turn	≥ 3	track-width correction	
Square closure	≥ 2	position/orientation closure error	

19.3. Ko analizēt

- Wheel slip nav redzams enkoderam — rats var griezties, bet šasija nepārvietojas atbilstoši.
- Gearbox backlash un grīdas berze var padarīt CW un CCW rezultātus atšķirīgus.
- Accurate wheel speed nenozīmē accurate pose; control un estimation ir dažādi sistēmas slāņi.
- Odometrijas kļūda uzkrājas ar nobraukto ceļu. Vēlāk SLAM/EKF palīdz, bet nekvalitatīva bāzes kalibrācija paliek problēma.

20. M5 — MPU6050 kalibrācija, asis un heading salīdzinājums



20.1. attēls. M5 Functional Tree — calibrated IMU measurements.

20.1. Teorija, ko stāstīt

Gyroscope mēra angular rate. Leņķi iegūst, integrējot ātrumu laikā. Tāpēc pat mazs pastāvīgs bias rada pieaugošu angle drift.

$$\text{yaw}(t) = \text{yaw}(0) + \text{integral}(\text{gz} * dt)$$

Static bias raksturo mērījuma vidējo nobīdi, bet standard deviation raksturo trokšņa izkliedi. Bias correction un filtering ir atšķirīgi uzdevumi.

20.2. Praktiskā secība

1. Ar fizisku pagriešanu nosaki X/Y/Z gyro axis un zīmju konvenciju.
2. Ar izslēgtiem motoriem ieraksti vismaz 60 s static telemetry.
3. Ar imu_bias_analysis.py aprēķini gx/gy/gz mean un stdev.
4. Uzliec runtime bias ar B komandu vai demonstrē firmware K kalibrāciju.

5. Izmēri 60 s yaw drift pirms un pēc correction.

6. Veic vismaz trīs 90° vai 360° pagriezienus un salīdzini physical angle, encoder theta un IMU integrated yaw.

```
python host_tools/serial_logger.py --port /dev/ttyUSB0 --seconds 60 --output results/M05_imu.csv
```

```
python host_tools/imu_bias_analysis.py results/M05_imu.csv
```

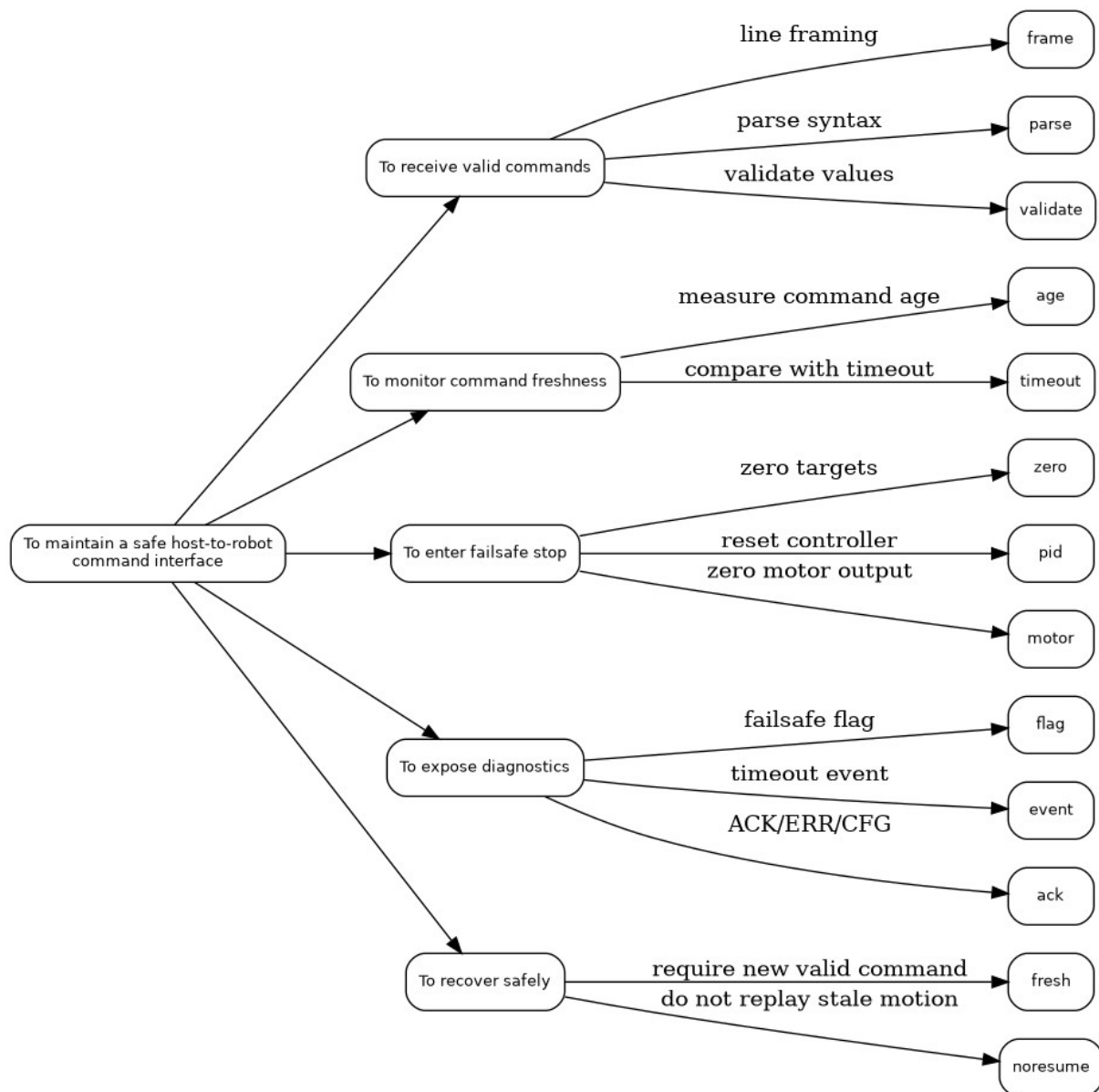
K # runtime gyro calibration; robot must remain stationary

Pierādījums	Expected	Actual	PASS/FAIL
Axis convention	documented		
Static dataset	>=60 s		
Bias mean	gx/gy/gz		
Noise stdev	gx/gy/gz		
Yaw drift before/after	measured		
Encoder vs IMU turns	>=3		

20.3. Svarīgs ierobežojums

Runtime K calibration tiek glabāta tikai RAM un pazūd pēc reset. Tas ir apzināti: students vispirms iemācās, ka kalibrācija ir identificējams parametrs ar pierādījumu. Persistent parameter storage tiek pievienots tikai tad, kad ir definēta konfigurācijas pārvaldība.

21. M6 — Serial protocols un command watchdog



21.1. attēls. M6 Functional Tree — command validation and failsafe stop.

21.1. Ko stāstīt

Autonoma robota host process, USB kabelis vai ROS 2 node var pazust. Tāpēc controller nedrīkst turpināt veco kustības komandu bezgalīgi. Application command watchdog mēra, cik vеча ir pēdējā validā velocity komanda.

```
if command_age_ms > COMMAND_TIMEOUT_MS:  
    target_v = 0  
    target_w = 0  
    reset PID  
    motor output = 0  
    failsafe = 1
```

Command watchdog nav tas pats, kas ESP32 task/interrupt watchdog. Pirmais kontrolē kustības komandu svaigumu; otrais palīdz atklāt iestrēgušu programmas izpildi. Drošības slāņi risina dažādus failure modes.

21.2. M6 testa secība

1. Saglabā ? konfigurācijas atbildi un COMMAND_TIMEOUT_MS.
2. Pārbaudi valid commands un ACK uz droši pacelta robota.
3. Nosūti malformed/unknown komandas un pierādi, ka tās nerada kustību.
4. Palaid watchdog_test.py un izmēri automātisko stop laiku.
5. Veic cable/process interruption testu.
6. Pēc reconnect pierādi, ka robots neatjauno veco kustību bez jaunas validas V komandas.

```
python host_tools/watchdog_test.py --port /dev/ttyUSB0 --timeout-ms 500
```

Failure injection	Expected response	Actual	PASS/FAIL
Host sender stops	timeout -> zero motion		
USB disconnect	timeout -> zero motion		
Malformed command	ignore/error, no unintended motion		
Telemetry consumer stops	local control continues		
MCU code hangs	application watchdog may not run; separate system watchdog needed		

22. M3–M6 kopējais acceptance gate

Pāreja uz M7 ROS 2 bridge notiek tikai tad, kad host-to-robot interface un fiziskā base ir izmērāmi stabila. ROS 2 nedrīkst izmantot, lai noslēptu neatrisinātu encoder, control vai failsafe problēmu.

Milestone	Obligātais evidence	Status
M3	PID step logs + metrics + left/right comparison + saturation explanation	
M4	Repeated straight/turn/square calibration + before/after parameters	
M5	Static IMU statistics + drift correction + encoder/gyro comparison	
M6	Measured stale-command failsafe + malformed-command test + recovery	

Digitālais protokols: 2WD_M3_M6_testi_un_kalibracija_v1_0.xlsx. Tas paredzēts kā vienots evidence summary, kamēr raw CSV faili paliek projekta results mapē.

23. Pasniedzēja M3–M6 sagatavošanas checklist

- References robotam M0–M6 iziet pašam pirms studentu darba.
- Saglabāt vienu known-good PID gain set, bet sākumā to nedot studentiem kā “pareizo atbildi”.
- Sagatavot drošu lifted-wheel fixture vai roller stand PID un watchdog testiem.
- Grīdā marķēt straight distance un turn reference; vienoties par chassis measurement reference point.
- IMU laboratorijā nodrošināt stingru, nekustīgu galda virsmu un motori OFF.
- Katram robotam lietot Robot ID, hardware revision, firmware commit un calibration date.
- M6 cable disconnect testu veikt tikai ar mazu speed un brīvu stop zonu / lifted fixture.
- Neļaut pāriet uz ROS 2, kamēr stale command stop nav fiziski demonstrēts.

24. M3–M6 papildu tehniskie avoti

Espressif Arduino-ESP32 LEDC API:

<https://docs.espressif.com/projects/arduino-esp32/en/latest/api/ledc.html>

Espressif Arduino-ESP32 Timer API:

<https://docs.espressif.com/projects/arduino-esp32/en/latest/api/timer.html>

Espressif ESP-IDF Watchdogs: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/wdts.html>

TDK InvenSense MPU-6000/MPU-6050 Product Specification and Register Map (legacy device documentation).

25. M7–M12 autonomijas slāņa mērķis

M7–M12 pārveido jau validētu M0–M6 mobilo bāzi par pilnu ROS 2 autonomās robotikas platformu. Šajā posmā galvenā disciplīna ir interfeisu, koordinātu sistēmu un pierādījumu kontrole: katru jauno slāni drīkst pievienot tikai pēc iepriekšējā slāņa PASS.

M6 safe serial base -> M7 ROS 2 -> M8 EKF -> M9 LiDAR -> M10 SLAM -> M11 Nav2 -> M12 Vision

Milestone	Galvenā funkcija	Ieejas gate	Galvenais rezultāts
M7	ROS 2 bridge un TF kontrakts	M6 PASS	/cmd_vel, raw odom, IMU, diagnostics
M8	Stāvokļa novērtēšana	M7 PASS	/odometry/filtered + viens odom->base_link TF
M9	2D LiDAR	M8 PASS	validēts /scan un laser extrinsic
M10	SLAM	M9 PASS	koherenta 2D karte + map->odom
M11	Nav2	M10 PASS	atkārtota autonomā goal navigation
M12	Machine Vision	M11 PASS	standarta 2D detection stream, neatkarīgs no navigācijas

26. M7 — ROS 2 bridge, topics un TF ownership

26.1. Teorija, ko stāstīt

ROS 2 šeit nav “maģiska programma virs robota”. Tas ir sadalītas sistēmas interfeisu slānis. Studentam jāspēj nosaukt katra topic publisher, subscriber, message type, frame un atbildības robeža.

Topic	Type	Publisher	Consumer
/cmd_vel	geometry_msgs/Twist	teleop / Nav2	serial bridge
/wheel/odometry_raw	nav_msgs/Odometry	serial bridge	EKF
/imu/data_raw	sensor_msgs/Imu	serial bridge	EKF
/odometry/filtered	nav_msgs/Odometry	robot_localization	Nav2 / diagnostics
/scan	sensor_msgs/LaserScan	LiDAR driver	SLAM / Nav2
/vision/detections	vision_msgs/Detection2DArray	vision node	application

TF noteikums: vienam transformam jābūt vienam autoritatīvam publisher. Hardware režīmā serial bridge publicē raw messages, bet EKF publicē odom -> base_link. URDF/robot_state_publisher publicē fiksētos sensoru transformus.

26.2. 90 min laboratorijas secība

1. Build un source ROS 2 workspace; palaist robot description + serial bridge + EKF.
2. Ar paceltiem riteņiem nosūtīt ļoti mazu /cmd_vel un pārbaudīt linear.x un angular.z zīmes.
3. Izmērīt /wheel/odometry_raw, /imu/data_raw un /odometry/filtered frekvences.
4. Pārbaudīt TF tree un pierādīt, ka odom -> base_link ir tikai viens īpašnieks.
5. Aktīvas mazas komandas laikā apturēt host bridge un izmērīt M6 watchdog stop.

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

6. Saglabāt evidence un aizpildīt M7 sheet darbgrāmatā.

```
ros2 topic hz /wheel/odometry_raw
ros2 topic hz /imu/data_raw
ros2 topic echo --once /odometry/filtered
```

27. M8 — EKF un lokālais stāvokļa novērtējums

M8 baseline apzināti ir vienkāršs: wheel odometry dod forward velocity v_x , bet IMU dod yaw rate v_{yaw} . Tas samazina risku vienu un to pašu informāciju bez izpratnes “iebarot” filtram vairākos veidos.

```
wheel odom: fuse vx
IMU: fuse vyaw
two_d_mode = true
world_frame = odom
```

27.1. Ko mērīt

- 30 s stationary drift un diagnostics
- atkārtoti straight 2 m braucieni
- $\pm 90^\circ$ vai 360° pagriezieni
- raw vs filtered distance/yaw
- TF ownership pēc EKF palaišanas

Covariance nav “smoothing knob”. To sākotnējām vērtībām jābūt sasaistītām ar M3–M5 mērījumu izkliedi un kļūdu raksturu. Nepamatoti maza covariance liek filtram pārmērīgi uzticēties konkrētam sensoram.

28. M9 — 2D LiDAR integrācija

M9 references interfeiss ir vendor-neutral: izvēlētais LiDAR driver publicē sensor_msgs/LaserScan uz /scan, un TF savieno tā frame ar base_link. Kursā BOM paredz RPLIDAR A1-class sensoru kā vienu praktisku references variantu.

28.1. LiDAR pirms SLAM

1. Stingri nostiprināt LiDAR un izmērīt xyz/rpy pret šasijas references punktu.
2. Pārbaudīt /scan frame_id, range_min/range_max un stabilu publicēšanas frekvenci.
3. RViz novietot sienu priekšā, pa kreisi un pa labi; pārbaudīt ģeometriju.
4. Pagriezt robotu uz vietas un pārliecināties, ka scan rotē ap paredzēto centru.
5. Pārbaudīt mehānisko mount un kabeļa strain relief.
6. Tikai pēc M9 PASS sākt SLAM.

```
python3 ros2_ws/src/two_wd_robot_bringup/scripts/scan_validator.py
```

29. M10 — 2D SLAM ar slam_toolbox

M10 PASS nav fakts, ka RViz parādījās occupancy grid. Karte ir jāpārbauda kā mērījumu produkts: mērogs, sienu dubultošanās, loop closure, pose stabilitāte un save/reload spēja.

Pārbaude	Ko pierāda	FAIL piemērs
Known distance vs map distance	kartes mērogu	2.00 m telpā kļūst par 2.30 m kartē
Closed loop	loop closure konsekvenci	atgriežoties sākumā sienas dubultojas
Straight walls	scan/odom/TF saskaņu	“spoku sienas” vai loki
Map save/reload	reproducējamību	karti nevar atkārtoti izmantot

29.1. Frame modelis

map -> odom -> base_link -> laser_frame

map ir globāli korigēts frame, odom ir lokāli nepārtraukts frame, kurš drīkst driftēt, bet nedrīkst lēkāt parastas kustības laikā.

30. M11 — Nav2 autonomā navigācija

Nav2 tiek palaists tikai pēc M10 PASS. Studentam jāsāk no konkrētās instalētās ROS 2/Nav2 versijas default configuration un tajā jāpārnes izmērītā footprint, ātruma robežas, sensoru topic un costmap izvēles. Kursa nav2_student_tuning.yaml ir worksheet, nevis universāla “kopē un brauc” konfigurācija.

30.1. Tuning secība

1. robot geometry + TF
2. sensor range/frame
3. velocity/acceleration limits
4. costmap obstacle marking + inflation
5. planner/controller behavior
6. Behavior Tree recovery behavior

30.2. Obligātie testi

- vismaz 5 goals no dažādiem sākuma heading
- final position/yaw error
- bloķēts ceļš
- recovery vai drošs failure
- costmap obstacle alignment
- command velocity maksimumu pārbaude pret fizisko robotu

31. M12 — Machine Vision paplašinājums

M12 pievieno kameru un perception pipeline, bet nepadara visu navigāciju atkarīgu no viena vision node. Baseline ir vienkāršs HSV color-target detector, jo students var redzēt threshold, mask, contour, area un bounding box. Tas nav pasniegts kā production AI detector.

USB camera -> sensor_msgs/Image -> cv_bridge/OpenCV -> vision_msgs/Detection2DArray

31.1. Baseline acceptance

- stable image topic and documented camera optical frame
- positive and negative scenes
- false-positive count
- Detection2DArray output
- debug image
- vision node stop does not break LiDAR navigation

31.2. Pāreja uz AI

Pēc baseline var aizstāt detector core ar ONNX/YOLO vai citu modeli, saglabājot to pašu ROS output contract. Tas studentam demonstrē sistēmas arhitektūras principu: algoritmu var nomainīt, ja interfeiss paliek stabils.

32. M7–M12 kopējais acceptance gate

Gate	Minimum evidence	Kas bloķē nākamo posmu
M7	ROS graph + TF + failsafe	topic/frame/sign vai duplicate TF problēma
M8	filtered odom + diagnostics	nepareizs sensor sign/covariance/TF
M9	/scan + RViz geometry	rotēts/mirrored scan vai brīvs mount
M10	loop-closed map + scale	doubled walls / severe map distortion
M11	goal success + recovery	unsafe limits / costmap mismatch
M12	vision metrics + independence	nestabils detector vai navigation coupling

Digitālais evidence summary: 2WD_M7_M12_acceptance_v1_0.xlsx. Raw rosbag, CSV, screenshots un map faili paliek projekta results/evidence mapē.

33. Pasniedzēja M7–M12 sagatavošanas checklist

- References robotam M0–M6 jābūt fiziski PASS pirms ROS 2 autonomijas nodarbībām.
- Pirms studentiem pārbaudīt ROS 2 workspace build, serial permissions un zināmu USB portu.
- Sagatavot vienu known-good TF/URDF un vienu apzināti kļūdainu variantu diagnostikas demonstrācijai.
- LiDAR laboratorijai sagatavot vienkāršu sienu/koridora testvidi ar zināmu izmēru.
- SLAM maršrutā obligāti iekļaut loop closure un vismaz vienu zināmu distances references mērījumu.
- Nav2 testam noteikt maksimālo ātrumu, stop zonu un fizisku motor power isolation iespēju.
- Vision laboratorijai sagatavot pozitīvos un negatīvos test scenes dažādos apgaismojumos.
- Saglabāt vienu known-good rosbag katram milestone, lai var mācīt offline debugging bez robotu rindas.

34. M7–M12 troubleshooting karte

Simptoms	Pārbaudi vispirms	Tad
Robots kustas, bet RViz nekustas	/wheel/odometry_raw rate/sign	EKF config un TF
EKF lēkā	IMU yaw sign/timestamp/covariance	sensoru duplicate fusion
Scan ir sāniski	laser_frame extrinsic	driver angle convention

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

Simptoms	Pārbaudi vispirms	Tad
SLAM dubulto sienas	scan+TF+odom	loop closure un kustības ātrums
Nav2 griežas uz vietas	footprint/costmap/TF	controller limits
Nav2 ignorē šķērsli	/scan costmap observation source	range/marking/clearing
Vision false positives	mask/threshold + scene	lighting/data/model
Vision node crash aptur navigāciju	architecture coupling	separate process/topic contract

35. Gala integrētā demonstrācija

1. Parādīt motoru emergency power isolation.
2. Parādīt M6 command watchdog stop.
3. Parādīt ROS graph un TF tree.
4. Parādīt /odometry/filtered un EKF diagnostics.
5. Parādīt /scan korekti saskaņotu RViz.
6. Parādīt map/localization stāvokli.
7. Nosūtīt vismaz vienu Nav2 goal un sasniegt to bez manuālas stūrēšanas.
8. Demonstrēt vienu bloķēta ceļa recovery vai drošu failure.
9. Parādīt /vision/detections un vienu false-positive/negative analīzi.
10. Noslēgumā parādīt evidence pack ar hardware revision, firmware commit, ROS workspace commit un calibration values.

36. M7–M12 tehniskie avoti

ROS 2 Lyrical: <https://docs.ros.org/en/lyrical/>

ROS 2 Lyrical release: <https://docs.ros.org/en/kilted/Releases/Release-Lyrical-Luth.html>

Nav2 first-time setup: https://docs.nav2.org/setup_guides/

Nav2 sensors: https://docs.nav2.org/setup_guides/sensors/setup_sensors_gz.html

Nav2 mapping/localization: https://docs.nav2.org/setup_guides/sensors/mapping_localization.html

Nav2 configuration: <https://docs.nav2.org/configuration/>

Nav2 tuning: <https://docs.nav2.org/tuning/>

Nav2 Behavior Trees: https://docs.nav2.org/behavior_trees/

usb_cam: https://docs.ros.org/en/rolling/p/usb_cam/

image_proc: https://docs.ros.org/en/kilted/p/image_proc/doc/components.html

vision_msgs: https://docs.ros.org/en/rolling/p/vision_msgs/

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

37. References platformas v3.0 statuss

Rokasgrāmata tagad nosedz pilnu M0–M12 vertikālo robotikas ķēdi. M0–M6 satur fiziskās bāzes, vadības, odometrijas, IMU un failsafe metodiku; M7–M12 satur ROS 2, state estimation, LiDAR, SLAM, Nav2 un Machine Vision integrācijas metodiku. Programmatūras sintakses un ROS-neatkarīgie testi var tikt automatizēti, bet gala PASS vienmēr piešķir pēc mērījuma uz konkrētās fiziskās platformas.

38. M13–M18 advanced robotikas un validācijas slānis

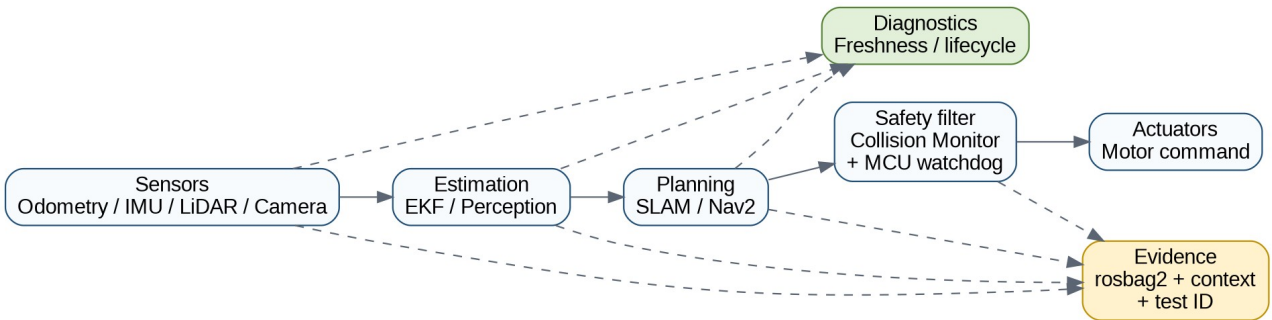
M0–M12 pierāda, ka robots var kustēties, mērīt, lokalizēties, kartēt, navigēt un publicēt Machine Vision rezultātus. M13–M18 pievieno nākamo inženiertehnisko slāni: reproducējamību, sim-real salīdzinājumu, kontrolētu kļūmju ievadi, programmatūras drošības slāni, sensoru fusion un prasību izsekojamību līdz testa pierādījumam.



38.1. attēls. M13–M18 quality-gate secība.

Posms	Funkcija	PASS pierādījums
M13	rosbag2 evidence	Bag info, topics/QoS, replay, test context
M14	Digital Twin / sim-real	Vienāds scenārijs, trajektorijas RMSE un gala kļūda
M15	Fault injection	Definēts fault ID, expected reaction, safe outcome, evidence
M16	Safety / diagnostics	Freshness, Collision Monitor, lifecycle uzvedība
M17	Camera + LiDAR fusion	Bearing/range cross-check ar negative/outlier testiem
M18	Capstone V&V	Requirement -> Test -> Evidence traceability + atkārtota misija

38.1. Pierādījumu arhitektūra



38.2. attēls. Datu, diagnostikas un evidence plūsma.

Būtiska prasība: evidence tiek vākti no tiem pašiem raw/estimated/planning/safety topics, kas piedalās robota darbībā. Ekrānattēls ir tikai papildus vizuāls pierādījums, nevis raw datu aizvietotājs.

39. M13 — rosbag2 evidence un reproducējams replay

M13 mērķis ir pārveidot “es redzēju, ka robots strādāja” par reproducējamu testu artefaktu. Students saglabā bag, software commit, robot ID, konfigurācijas kontekstu un testa ID.

39.1. Minimālais ierakstāmo topic komplekts

- /tf un /tf_static
- /wheel/odometry_raw

- /odometry/filtered
- /imu/data_raw
- /scan
- /cmd_vel
- /diagnostics
- /detections vai kameras output, ja tas ir konkrētā testa mērķis

39.2. Praktiska secība

```
mkdir -p evidence/M13
```

```
ros2 bag record /tf /tf_static /wheel/odometry_raw /odometry/filtered /imu/data_raw /scan /cmd_vel /diagnostics -o evidence/M13/baseline_run
```

```
ros2 bag info evidence/M13/baseline_run
```

```
ros2 bag play evidence/M13/baseline_run
```

Sensoru QoS var atšķirties no default reliable profila. Projekta

`advanced\_validation/config/qos\_overrides.yaml` demonstrē explicit rosbag2 QoS override pieeju. PASS nav “mape eksistē”; PASS nozīmē, ka obligātie topics ir ierakstīti un replay laikā consumers saņem datus.

39.3. M13 quality gate

Pārbaude	Expected	Evidence
Bag metadata	duration > 0 un obligātie topics redzami	ros2 bag info output
Replay	pipeline saņem replay datus	terminal log / analysis output
QoS	sensor topic nav klusi pazudis	QoS config + message counts
Context	evidence piesaistīts versijai	test_context.json / commit ID

40. M14 — Digital Twin un sim-real salīdzinājums

Šajā kursā Digital Twin tiek lietots praktiskā mācību nozīmē: simulācijas modelis satur izmērītus fiziskā robota parametrus, tam tiek dots tas pats testu profils, un simulācijas rezultātu salīdzina ar fizisko robotu ar vienādi definētām metrikām.

40.1. Salīdzināmās metrikas

- trajektorijas XY RMSE
- gala pozīcijas kļūda
- gala yaw kļūda
- nobrauktais ceļš
- commanded vs measured velocity profila atšķirība
- testa izpildes laiks un kavējumi

40.2. Kalibrācijas disciplīna

1. No M2–M4 mērījumiem ievada wheel radius un track width.
2. Simulācijā izmanto to pašu command profile.
3. Fiziskajā robotā izpilda vienu un to pašu scenāriju vismaz 3 reizes.
4. Kalibrācijas scenārijus nodala no validācijas scenārija.
5. `sim\_real\_metrics.py` aprēķina RMSE un gala kļūdas.

6. Students paskaidro, kāpēc sim un real nekad nebūs pilnīgi identiski.

41. M15 — kontrolēts fault injection un resilience

Fault injection tiek veikts programmatūras/datu plūsmas līmenī. Nav atļauts apzināti īsslēgt barošanu vai turēt motoru stall režīmā. Visus fault testus sāk ar samazinātu ātrumu, tukšu testu zonu un fizisku motoru barošanas atslēgšanas iespēju.

Fault ID	Injekcija	Expected sistēmas reakcija
F01	/scan pazūd	diagnostics kļūda; navigācija neizmanto novecojušu scan kā normālu
F02	NaN sektors /scan	node nekrīt; invalid rays tiek ignorēti
F03	scan delay / stale	freshness kļūda noteiktā timeout
F04	serial bridge process stop	M6 command watchdog aptur motorus
F05	bloķēts ceļš	Nav2 recovery vai controlled failure, nevis sadursme
F06	vision node/detection pazūd	LiDAR/Nav2 pamatdrošība turpina darboties neatkarīgi

41.1. Fault testā obligāti jāmēra

- fault injection laiks
- fault detection laiks
- safe stop / slowdown laiks, ja attiecas
- recovery veids
- vai nepieciešama cilvēka iejaukšanās
- evidence bag un log

42. M16 — Collision Monitor, diagnostics un lifecycle

M16 apvieno trīs dažādus aizsardzības mehānismus: MCU stale-command watchdog, ROS health/freshness diagnostics un Nav2 Collision Monitor kā papildu kustības komandas filtrēšanas slāni. Nevienu no tiem šajā mācību platformā neuzskata par sertificētu funkcionālās drošības risinājumu.

42.1. Collision Monitor baseline

Projekta `advanced\_validation/config/collision\_monitor.yaml` satur priekšējo StopZone un lielāku SlowdownZone. Pirms kustības pasniedzējs/studenti RViz vai topic līmenī pārbauda, ka zonas atrodas pareizajā base_link koordinātu sistēmā.

42.2. Freshness monitor

Plūsma	Sākuma timeout	Kļūmes interpretācija
/odometry/filtered	0.50 s	state estimation nav atjaunināta
/scan	0.50 s	nav svaiga šķēršļu uztveres avota
/camera/image_raw	1.00 s	vision degradēts; navigācijas pamatdrošību nedrīkst padarīt atkarīgu no šī viena signāla

Timeout sākuma vērtības ir teaching thresholds. Reālajā platformā tās pamato ar izmērīto publicēšanas frekvenci, jitter un drošības rezervi.

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

43. M17 — Camera + LiDAR perception fusion

M17 mērķis nav “AI automātiski saprot pasauli”, bet izskaidrojams divu sensoru cross-check. Kameron bounding box centra horizontālā pozīcija tiek pārvērsta bearing leņķī, pēc tam LiDAR leņķiskajā logā izmanto valid ranges mediānu.

$$\text{bearing} = (\text{cx} / \text{image_width} - 0.5) * \text{camera_HFOV}$$

43.1. Acceptance scenāriji

- target kadrs ar skaidru LiDAR atbalstu
- target kadrs bez valid LiDAR rays
- negative scene bez target
- outlier LiDAR ray starp normāliem paraugiem
- target pie FOV malas
- vision node restart

Fusion output nedrīkst tikt pasludināts par safety-rated occupancy avotu. 2D LiDAR nevar noteikt objekta augstumu, bet kamera var kļūdīties apgaismojuma, blur vai nepareizas klases dēļ.

44. M18 — Capstone validation, traceability un release

M18 noslēdz robotu kā verificējamu sistēmu. Students izveido requirements-to-test-to-evidence matricu un final mission vairs netiek pieņemts pēc “viena veiksmīga video”.

44.1. Obligātais mission profile

1. Robotu ieslēdz safe state un parāda motor power isolation.
2. Pārbauda M6 watchdog.
3. Pārbauda ROS graph, TF un diagnostics.
4. Saņem autonomu goal un uzsāk navigāciju.
5. Apbrauc statisku šķērsli.
6. Collision Monitor stop/slowdown zonā reaģē paredzami.
7. Kontrolēta blockage gadījumā izpilda recovery vai controlled failure.
8. Vision target tiek sasaistīts ar LiDAR range, ja ir valid geometry.
9. Visa būtiskā datu plūsma tiek saglabāta rosbag2 evidence.
10. Misijas beigās motors droši apstājas un evidence tiek sasaistīts ar release versiju.

44.2. Requirements-to-evidence princips

ID	Prasība	Tests	Evidence
R001	Stale command -> non-driving state	T_M06_WATCHDOG	watchdog log / bag
R002	Vienam owner jāpublicē odom->base_link	T_M07_TF_OWNER	TF tree / topic evidence
R003	Collision zone samazina vai aptur command	T_M16_COLLISION_ZONE	cmd_vel in/out + scan bag

ID	Prasība	Tests	Evidence
R004	Critical sensor stale ir redzams diagnostics	T_M16_FRESHNESS	DiagnosticArray evidence
R005	Final mission ir traceable bag artefakts	T_M18_EVIDENCE	bag + context + commit

45. M13–M18 pasniedzēja sagatavošanas checklist

- Sagatavot known-good bag ar /tf, odom, IMU un scan offline demonstrācijai.
- Sagatavot vienu sim un vienu real trajektorijas CSV sim-real aprēķinam.
- Fault injection testos noteikt max ātrumu un clear test zone.
- Pirms Collision Monitor kustības testa vizuāli validēt polygon frame un topic remap.
- Izmērīt tipisko topic rate/jitter, pirms noteikt freshness timeout.
- Fusion laboratorijai izmērīt kameras horizontālo FOV vai izmantot dokumentētu/calibrated vērtību.
- Final demo sagatavot prasību sarakstu, evidence naming convention un reviewer lauku.

46. M13–M18 troubleshooting karte

Simptoms	Pārbaudi vispirms	Tad
Bag ir, bet /scan tukšs	ros2 topic info --verbose /scan; QoS	QoS override; driver rate; bag info
Replay laikā TF "lūkā"	vai nav divi TF owner	izslēdz duplicate publisher; pārbaudi /tf_static
Sim un real RMSE milzīgs	vienāds frame/time sampling/units	wheel radius, track, delays, slip
Fault injector rada crash	message validity un mode	samazini fault scope; log raw/modified
Collision Monitor nestopē	cmd_vel remap un scan source	polygon frame, source timeout, lifecycle state
Fusion range nepareizā pusē	camera optical frame un HFOV sign	LiDAR angle_min/increment; extrinsic alignment
Traceability checker fail	requirement/test ID neatbilst	evidence path, PASS status, mandatory rows

47. M13–M18 tehniskie avoti un versiju piezīme

ROS 2 Lyrical Luth release: <https://docs.ros.org/en/kilted/Releases/Release-Lyrical-Luth.html>

ROS 2 rosbag2 QoS overrides:

https://docs.ros.org/en/ros2_documentation/lyrical/How-To-Guides/Overriding-QoS-Policies-For-Recording-And-Playback.html

Gazebo / ROS compatibility: https://gazebo-sim.org/docs/jetty/ros_installation/

Gazebo ROS 2 Simulation Interfaces: https://gazebo-sim.org/docs/latest/ros2_sim_interfaces/

Nav2 Collision Monitor: <https://docs.nav2.org/configuration/packages/configuring-collision-monitor.html>

Nav2 Lifecycle Manager: <https://docs.nav2.org/configuration/packages/configuring-lifecycle.html>

References vide šai rokasgrāmatas versijai: ROS 2 Lyrical + Gazebo Jetty. Klasē drīkst saglabāt Jazzy/Harmonic, ja infrastruktūra jau ir uz tās standartizēta; tad komandas/parametrus pārbauda pret konkrētās instalētās versijas dokumentāciju.

48. M0–M18 checkpoint (v4.0)

v4.0 checkpoint nosedza M0–M18. v5.0 turpina ar M19–M24 industriālo paplašinājumu: embedded timing, CAN/TWAI, CI/Docker, depth/3D perception, multi-robot/fleet un release engineering.

49. M19 — Embedded real-time un micro-ROS migrācijas robeža

M19 mērķis ir pāriet no “cikls parasti izpildās pietiekami ātri” uz mērāmu timing budžetu. Real-time šajā kursā nozīmē definētu periodu, deadline, jitter un pierādījumu, cik bieži deadline tiek pārkāpts.

49.1. Obligātais references ceļš

- Esošais ESP32 serial protocol + M6 command watchdog paliek kustības references drošības bāze.
- Control loop sākuma periods: 10 ms; faktiskā vērtība jāapstiprina ar mērījumiem.
- Students ieraksta vismaz 1000 ciklu timestamp datus un aprēķina mean, min, max, p99 un deadline miss ratio.
- Overload testu sāk ar motor power izolētu vai robotu droši nostiprinātu.

49.2. micro-ROS versiju disciplīna

2026-08-19 micro_ros_platformio oficiāli dokumentē Humble, Jazzy, Kilted un Rolling; Lyrical nav supported values sarakstā. Tāpēc šī rokasgrāmata nepasļudina Lyrical micro-ROS integrāciju par pārbaudītu. Optional migration laboratorija izmanto Kilted profilu, bet Lyrical references robots saglabā serial bridge, kamēr konkrēta client/agent/distro ķēde nav atsevišķi validēta.

```
board_microros_distro = kilted
```

```
board_microros_transport = serial
```

49.3. M19 acceptance

Metrika	Expected / note	Evidence
Control period	definēts pirms testa	timing_budget.yaml
p99	zem mācību threshold vai tehniski pamatots	timing CSV + rt_budget_check.py
Deadline misses	mērīti, nevis minēti	summary
Overload behavior	nav nekontrolētas kustības	test log/video
micro-ROS	optional Kilted migration only	version note

50. M20 — CAN/TWAI komunikācija

ESP32 TWAI kontrolieris ļauj ieviest CAN tipa kopni, bet kontrolierim nav integrēta physical-layer transceivera. Reālai kopnei nepieciešams ārējs transceiveris. M20 māca ne tikai frame nosūtīšanu, bet heartbeat, sequence, bus-off un safe-state reakciju.

50.1. References frame contract

```
0x100 wheel command: left_mm_s:int16, right_mm_s:int16, seq:uint8, flags:uint8, timeout_ms:uint16
```

```
0x110 wheel status
```

```
0x120 diagnostics / heartbeat
```

- Classic CAN command frame ir 8 baiti.
- Sequence counter palīdz konstatēt izlaistas/stale application komandas.
- Application watchdog paliek obligāts arī tad, ja CAN nodrošina frame CRC un link-layer error detection.

50.2. Fiziskā kopne

- 120 ohm termination — abos kopnes fiziskajos galos, nevis pie katra mezgla.
- Pirmais tests bez motor enable: tikai heartbeat un diagnostics.
- Motor power un logic power topoloģija saglabā M0 principus.
- Bus-off vai heartbeat timeout gadījumā kustības komanda tiek anulēta un robots pāriet stop/safe state.

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

51. M21 — CI, Docker un reproducējama izstrādes vide

M21 galvenais jautājums: vai projektu var pārbaudīt no tīra source snapshot, nevis tikai tajā datorā, kur tas tika izstrādāts?

51.1. CI pipeline minimums

1. Clean checkout.
2. Pure-Python testi.
3. Native C++ control-core tests.
4. Package verification un failu sintakses pārbaude.
5. Docker image build.
6. ROS workspace build/test, kad references ROS runner ir pieejams.
7. Saglabāts CI log vai artefakts.

51.2. Docker references vide

FROM osrf/ros:lyrical-desktop

ROS 2 Lyrical dokumentācijā šis image tags tiek izmantots Docker darba piemērā. Build cache drīkst izmantot ātrumam, bet release pirms M24 jāspēj pārbaudīt arī no clean stāvokļa.

51.3. Secrets un reproducējamība

- API keys, paroles un tokeni nedrīkst būt source repo plaintext formā.
- Konfigurācija un kalibrācija tiek versionēta vai skaidri piesaistīta release artefaktam.
- Container image neaizstāj hardware acceptance — tas pierāda programmatūras vides reproducējamību.

52. M22 — Depth kamera un 3D perception

M22 ir vendor-neitrāls. Pirms konkrētas RGB-D kameras draivera students pierāda depth mērvienības, intrinsics un camera optical frame ģeometriju.

52.1. Pinhole deprojection

$$X = (u - cx) * Z / fx$$

$$Y = (v - cy) * Z / fy$$

$$Z = \text{depth}$$

Depth = 0, NaN, infinity vai ārpus definētā range tiek uzskatīts par invalid measurement, nevis par šķērslī ar nulles attālumu.

52.2. ROS datu kontrakts

- Depth image: sensor_msgs/Image.
- Intrinsics: sensor_msgs/CameraInfo.
- 3D punkts: sensor_msgs/PointCloud2.
- ROS image_pipeline/depth_image_proc var izmantot depth image -> point cloud pipeline, bet pure-Python depth_geometry.py paliek matemātikas etalons bez ROS atkarībām.

52.3. M22 acceptance

Tests	Mērījums	PASS princips
1 m references plakne	range/scale error	$\leq$ teaching threshold vai pamatots
Invalid depth	invalid ratio	izmērīts un dokumentēts
Point cloud	point count + nearest obstacle	ģeometrija loģiska
TF	camera optical frame	vizuāli/matemātiski saskaņots
Rate	Hz + jitter note	pietiek konkrētajai funkcijai

53. M23 — Multi-robot un fleet

M23 atdala namespace, TF frame uniqueness un ROS_DOMAIN_ID. Tie risina dažādas problēmas, tāpēc tos nedrīkst lietot kā savstarpējus aizvietotājus.

53.1. References fleet naming

/robot1/... frames: robot1/base_link, robot1/odom

/robot2/... frames: robot2/base_link, robot2/odom

- Vienas fleet roboti paliek vienā ROS domain, lai var savstarpēji discover/komunicēt.
- Atšķirīgus ROS_DOMAIN_ID izmanto neatkarīgu klases grupu/tīklu izolācijai.
- Fleet supervisor piešķir high-level goal; lokālais watchdog, Collision Monitor un motor control paliek konkrētajā robotā.

53.2. Fleet allocator

References fleet_allocator.py izmanto vienkāršu izmaksu funkciju pēc attāluma, availability un battery threshold. Tas ir izskaidrojams baseline, nevis optimāls fleet scheduling algoritms.

53.3. Failure scenāriji

- Viens robots pazūd no tīkla.
- Goal tiek piešķirts unavailable robotam.
- Diviem robotiem kļūdaini sakrīt TF frame names.
- Lokālā safety sistēma aptur robotu neatkarīgi no fleet supervisor.

54. M24 — Industrial release un handover

M24 nepievieno jaunu autonomijas algoritmu. Tas nosaka, vai konkrēto robota programmatūras/aparatūras stāvokli drīkst saukt par identificētu release.

54.1. Release artefakti

- semantic version / release ID;
- source commit vai snapshot identifikators;
- hardware revision un firmware versija;
- ROS distro + konfigurācijas/kalibrācijas komplekts;
- SHA-256 file manifest;
- M0–M24 acceptance status;
- risk register un known limitations;
- rollback/restore procedūra;
- reviewer RELEASE / HOLD lēmums.

54.2. Release lēmums

Ja mandatory gate nav PASS vai evidence nav identificējams, lēmums ir HOLD. "Demo nostrādāja vienreiz" nav release kritērijs.

Requirement -> Test -> Evidence -> Version -> Reviewer decision

55. M19-M24 pasniedzēja sagatavošanas checklist

Pirms laboratorijas	Jā/Nē	Piezīme
M19: sagatavots timing CSV piemērs un motor power isolation	[]	
M20: CAN transceiveri, termination un can0/USB-CAN pieejams	[]	
M21: Docker/CI pieejams vai sagatavots offline piemērs	[]	
M22: depth dataset vai RGB-D kamera + known 1 m target	[]	
M23: divi namespace profili vai multi-robot simulācija	[]	
M24: release checklist, risk register un reviewer forma	[]	
Visiem: evidence naming convention un droša testu zona	[]	

56. M19-M24 troubleshooting karte

Simptoms	Pārbaudi vispirms	Tipisks cēlonis
p99 ir liels, mean normāls	background tasks / logging	jitter nav redzams vidējā vērtībā
CAN frames neredzami	termination, bitrate, transceiver power	physical layer / config
CAN ir, bet motors turpina stale komandu	application watchdog	heartbeat nav sasaistīts ar safe state
Docker build strādā, CI nē	paths, permissions, clean checkout	lokāls neversionēts fails
Depth scale x1000	units mm vs m	nepareiza depth mērvienība
Point cloud pagriezts	optical frame / TF	axis convention kļūda
robot1 redz robot2 TF kā savu	frame names / namespace	TF nav unikāls
Release manifest mainās bez source izmaiņām	generated caches/logs	manifestā iekļauti volatile faili

57. M19-M24 tehniskie avoti un references statuss

ROS 2 Lyrical Ubuntu: <https://docs.ros.org/en/lyrical/Installation/Alternatives/Ubuntu-Install-Binary.html>

ROS 2 Lyrical Docker: <https://docs.ros.org/en/lyrical/How-To-Guides/Run-2-nodes-in-single-or-separate-docker-containers.html>

ROS_DOMAIN_ID: <https://docs.ros.org/en/lyrical/Concepts/Intermediate/About-Domain-ID.html>

ROS namespaces: <https://docs.ros.org/en/lyrical/Tutorials/Intermediate/Launch/Creating-Launch-Files.html>

micro-ROS PlatformIO: https://github.com/micro-ROS/micro_ros_platformio

ESP32 TWAI: <https://docs.espressif.com/projects/esp-idf/en/v5.2/esp32/api-reference/peripherals/twai.html>

Docker GitHub Actions cache: <https://docs.docker.com/build/ci/github-actions/cache/>

ROS image_pipeline: https://docs.ros.org/en/rolling/p/image_pipeline/

Nav2 multi-robot namespace migration note: <https://docs.nav2.org/migration/Jazzy.html>

References robotikas host vide paliek ROS 2 Lyrical uz Ubuntu 26.04. micro-ROS M19 optional migrācijas profils apzināti ir Kilted, jo pašreizējais micro_ros_platformio supported distro saraksts Lyrical neuzrāda. M24 release dokumentācijā šo versiju robežu ieraksta kā known integration constraint.

58. References platformas v5.0 statuss

Rokasgrāmata tagad nosedz M0–M24: no barošanas, motoriem un enkoderiem līdz PID, odometrijai, IMU, ROS 2, LiDAR, SLAM, Nav2, Machine Vision, evidence/validation, embedded timing, CAN/TWAI, CI/Docker, depth/3D perception, multi-robot/fleet un industrial release/handover. Katrs nākamais slānis ir quality gate virs iepriekš pierādīta pamata.

59. M25 — manipulatora modelis, URDF/SRDF un controller contract

M25 sāk mobile manipulator paplašinājumu. Šajā posmā students vēl neveic pick-and-place un neieslēdz AI. Mērķis ir pierādīt pašu manipulatora kinemātisko modeli, joint limitus, TF ķēdi un controller robežu. References modelis EduArm6 ir vendor-neutral, simulation-first 6-DOF mācību roka ar divpirkstu gripperi; tas nav konkrēta komerciāla robota mehāniskais projekts un nav automātiski uzskatāms par drošu fiziskai izgatavošanai.

59.1. Kāpēc M25–M30 references workspace ir ROS 2 Kilted

Movelt 2 oficiālajā repozitorijā stabilās release branches pašlaik ir Humble, Jazzy un Kilted. Lyrical nav uzrādīta kā stabila Movelt release līnija. Tāpēc M25–M30 manipulatora trasei izmantojam Kilted references workspace. M30 pirms pilnas integrācijas pieprasa 2WD custom ROS pakotņu Kilted compatibility gate; līdz tam bāzes M0–M24 un manipulatora M25–M29 testus drīkst veikt atsevišķi.

59.2. Modelim obligāti jāatbild uz četriem jautājumiem

- Kāda ir kinemātiskā ķēde no base_link līdz tool0?
- Kāda ir katras ass rotācijas/translācijas ass un pieļaujamais diapazons?
- Kas ir planning group, end-effector un named states home/stow?
- Kā trajectory komanda nonāk controller slānī un kā atgriežas joint state feedback?

Elements	References prasība	Pierādījums
URDF/Xacro	6 bounded revolute joints + gripper + collision geometry	URDF contract test + RViz
SRDF	arm, gripper, end-effector, home/stow/open/closed	Movelt config / RViz
ros2_control	joint state + arm trajectory + gripper controller contract	controller list + topic/action evidence
TF	viena nepārtraukta ķēde base_link -> tool0	tf tree / frame inspection

59.3. M25 acceptance

M25 ir PASS tikai tad, ja visi seši arm joints ir identificēti ar limitiem, home un stow nav acīmredzamā self-collision stāvoklī, TF ķēde nav pārrauta un students spēj izskaidrot, kāpēc programmatūras modelis vēl nepierāda fiziskas rokas mehānisko drošību.

60. M26 — Movelt 2 motion planning un Planning Scene

M26 mērķis ir iemācīt atšķirību starp “mērķis eksistē” un “mērķi drīkst sasniegt ar derīgu trajektoriju”. Movelt MoveGroupInterface ļauj definēt joint vai pose mērķus un izveidot motion plan, savukārt Planning Scene satur robotu, vidi, collision objektus un attached objektus.

60.1. Obligātā testu secība

1. Palaid tikai simulācijas/references Movelt vidi un pārbaudi joint_states, planning frame un TF.
2. Izpildi joint-space plānu uz home un saglabā planning result.
3. Izpildi pose goal sasniedzamā vietā.
4. Planning Scene pievieno galda collision object.
5. Apzināti definē mērķi, kuru bloķē collision object, un pierādi, ka plāns tiek noraidīts.

6. Atkārtο vismaz piecus derīgus planning mēģinājumus un salīdzini planning time/path raksturu.

60.2. Kāpēc deliberate negative test ir obligāts

Ja students parāda tikai veiksmīgu kustību, nav pierādīts, ka collision modelis tiešām darbojas. Tāpēc M26 acceptance satur vismaz vienu apzināti nederīgu mērķi, kuru plānotājam jānoraida. Tas ievieš validācijas domāšanu: sistēma jāpārbauda gan ar pozitīvu, gan negatīvu scenāriju.

Tests	Expected	Evidence
Joint goal	Plan success	trajectory/plan log
Pose goal	Plan success	planning result + RViz
Collision-negative goal	Plan failure/rejection	failure code/log + scene
Out-of-limit joint	Rejected before unsafe execution	request/result evidence

61. M27 — Pick & Place ar MoveIt Task Constructor

Kursa references pick-and-place ceļš ir MoveIt Task Constructor (MTC), nevis vecais vienas funkcijas pick/place piemērs. MTC sadala sarežģītu uzdevumu stages, tāpēc students var redzēt, kur tieši planning pipeline izdodas vai neizdodas.

61.1. References stage ķēde

```
CurrentState -> Open -> Connect -> GenerateGrasp -> ComputeIK -> Approach  
-> Close -> Attach -> Lift -> ConnectToPlace -> Lower -> Open -> Detach  
-> Retreat -> ReturnHome
```

Šī ķēde kļūst par Functional Tree praktisko ekvivalentu: katrs stage ir pārbaudāma funkcija ar ieeju, rezultātu un failure mode.

61.2. Ko students dokumentē

- Planning Scene stāvokli un objekta frame.
- Grasp pose, approach direction un retreat direction.
- MTC stage tree ar veiksmīgajiem/neveiksmīgajiem zariem.
- Vismaz piecu atkārtotu mēģinājumu rezultātu.
- Failure klasifikāciju: perception, IK, collision, grasp, controller/execution.

61.3. M27 acceptance

Simulācijas references testā pēc sākotnējās konfigurācijas mērķis ir vismaz četri veiksmīgi mēģinājumi no pieciem identiskā scenārijā. Fiziskā references kritērijus pielāgo konkrētajam hardware, taču vienmēr saglabā zemu ātrumu, vieglu testobjektu un skaidru stop procedūru.

62. M28 — Force/Torque, kontakta noteikšana un MoveIt Servo robeža

M28 ievieš kontaktu kā izmērāmu signālu, nevis subjektīvu “robots piespiedās”. ros2_control Force Torque Sensor Broadcaster publicē WrenchStamped datus; references kursā students analizē bias, noise, frame zīmes un threshold reakciju.

62.1. References contact gate

```
CLEAR -> SOFT_CONTACT -> STOP
```

Piemēra soft-contact un stop robežas pakotnē ir tikai mācību sākuma vērtības. Tās nedrīkst mehāniski pārnest uz citu sensoru, citu instrumentu vai citu robotu bez kalibrācijas un riska analīzes.

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

Mērījums	Kāpēc vajadzīgs	Tipiska kļūda
Static bias	nulles nobīdes noteikšanai	threshold tiek skaitīts no nepareizas nulles
Noise/std	false-trigger riska novērtēšanai	sliexsnis zem trokšņa pīķiem
Frame/sign	pareizas spēka interpretācijas dēļ	+Z/-Z sajaukums
Stale-data	sensoru pazušanas reakcijai	vecs wrench tiek uzskatīts par aktuālu

62.2. MoveIt Servo

MoveIt Servo var pieņemt joint velocity, end-effector velocity vai pose tipa komandas un nodrošina arī singularity/collision handling iespējas. Kursā Servo drīkst lietot tikai pēc validēta URDF/SRDF, controller, joint feedback un collision modeļa. Tas nav attaisnojums apiet M25–M27 quality gates.

62.3. Safety robeža

Programmatūras force threshold un MoveIt Servo collision checking nav sertificēta emergency-stop sistēma. Fiziskai rokai nepieciešams neatkarīgs veids, kā atslēgt vai droši apturēt piedziņas enerģiju atbilstoši konkrētās sistēmas riskam.

63. M29 — AI-assisted object handling un grasp kandidātu ranking

M29 AI loma ir apzināti ierobežota. AI/perception ģenerē objekta kandidātu un novērtējumu, bet gala execution lēmumu pieņem deterministiski gates: depth validitāte, sasniedzamība, planning scene, collision checking, MTC feasibility un mission state.

63.1. References pipeline

RGB/depth -> detection -> 3D hypothesis -> candidate score
-> reachability/geometry gate -> Planning Scene -> MTC -> execution

63.2. Obligātie negative tests

Scenārijs	AI kandidāts	Expected sistēmas lēmums
High confidence, invalid depth	var būt augsts confidence	REJECT
Objekts ārpus reach	var būt pareizi atpazīts	REJECT
Neatļauta klase	var būt pareizi atpazīta	REJECT
Objekts collision zonā	labs 3D kandidāts	PLANNER REJECT
Perception pazūd pirms execution	iepriekš bija derīgs	HOLD/REPLAN

63.3. Ko AI nedrīkst darīt

References arhitektūrā AI node nedod tiešu motora PWM, joint effort vai nekontrolētu trajectory command. Tā publicē kandidātu vai target hypothesis. Šī robeža padara kļūdu analizējamu un saglabā atbildību par limitu/collision/safety pārbaudi atsevišķos deterministiskos slāņos.

64. M30 — mobile manipulator integrācija un gala misija

M30 savieno 2WD autonomo bāzi ar manipulatoru vienā misijā. Galvenais references princips: kamēr roka aktīvi manipulē, bāzes normāla kustība ir bloķēta. Coordinated base+arm motion ir atsevišķa advanced tēma un netiek automātiski uzskatīta par validētu tikai tāpēc, ka abi subsistēmas atsevišķi darbojas.

64.1. Kilted compatibility gate pirms integrācijas

Gate	PASS nosacījums
Base custom packages	kompilējas Kilted workspace
M7 contract	cmd_vel/odometry/IMU bridge smoke test
TF	nav dubultu transformu publisheru
M8–M12	EKF, LiDAR, Nav2, camera/depth smoke tests
Manipulator	Movelt demo + controller contract PASS

64.2. References mission state machine

IDLE -> NAVIGATE -> BASE_LOCK -> PERCEIVE -> PLAN -> EXECUTE_PICK
-> VERIFY_GRASP -> STOW -> BASE_UNLOCK -> NAVIGATE/PLACE -> DONE
fault / timeout / estop -> FAIL

64.3. Obligātie fault scenarios

- Perception timeout vai target pazūd.
- Movelt planning failure.
- Force STOP threshold.
- Grasp verification failure.
- Nav2 failure pirms base lock.
- Communication loss.

Katram fault scenārijam jābūt definētai safe reaction un saglabātam evidence. “Mēs nospiedām reset un turpinājām” nav validācijas rezultāts.

64.4. M30 gala acceptance

References mērķis: trīs atkārtotas pilnas misijas bez neparedzētas bāzes kustības manipulācijas laikā, ar requirements-to-test-to-evidence matricu, zināmo ierobežojumu sarakstu un reviewer RELEASE/HOLD lēmumu.

64.5. Fiziskās stabilitātes gate

Pirms reālas rokas montāžas uz 2WD bāzes jānovērtē kopējā masa, rokas masa, payload, reach, mounting plate un sliktākā masas centra nobīde. Pakotnē iekļautais static tipping-moment piemērs ir tikai mācību konservatīvs aprēķins; dinamiska kustība, paātrinājums, grīdas slīpums un riteņu kontakts var samazināt stabilitāti. Fiziskas rokas izvēle ir PASS tikai pēc konkrētā hardware datu dokumentēšanas un atbilstoša stabilitātes/strukturālā novērtējuma.

65. M25–M30 pasniedzēja sagatavošanas checklist

Pirms laboratorijas	Jā / Nē
Kilted references workspace snapshot/version ir dokumentēts	[]
Movelt/ros2_control pakotnes un demo starts pārbaudīts	[]
Simulācijas robot modelis atveras bez TF kļūdām	[]
Fiziskai rokai definēts zems speed/acceleration limits	[]
Testa zona brīva, objekti viegli un neasi	[]
Ir fizisks motor-power isolation/stop risinājums	[]
Studentiem ir acceptance workbook un evidence mapes	[]

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

Pirms laboratorijas	Jā / Nē
M30 base-lock behavior pārbaudāms neatkarīgi no AI	[]

66. M25–M30 troubleshooting karte

Simptoms	Pārbaudi vispirms	Tālāk
RViz roka “salūzusi”	TF/joint names/URDF axis	SRDF + fixed joints + joint states
Movelt neplāno	start state, limits, collision	IK/reachability/planner config
Trajektorija plānojas, bet nekustas	controller/action mapping	ros2_control interfaces/hardware
MTC apstājas vienā stage	stage failure iemesls	grasp pose/IK/collision/properties
Force trigger uzreiz	bias/frame/noise	filter + threshold + stale data
AI izvēlas sliktu objektu	depth/reachability/class gate	score weights + planning scene
Bāze kustas grasp laikā	mission state/base-lock	lower-level interlock/watchdog
M30 pakotnes nestrādā kopā	ROS distro/API mismatch	Kilted migration gate, dependencies

67. M25–M30 tehniskie avoti un references statuss

Pārbaudīts 2026-08-19. References materiāli balstīti uz oficiālo Movelt 2 un ros2_control dokumentāciju. Kurša arhīvā avoti ir arī saglabāti kā plain-text URL, lai pasniedzējs var atkārtoti pārbaudīt versiju statusu.

- Movelt 2 repository / stable release branches — <https://github.com/moveit/moveit2>
- Move Group C++ Interface — https://moveit.picknik.ai/main/doc/examples/move_group_interface/move_group_interface_tutorial.html
- Planning Around Objects — https://moveit.picknik.ai/main/doc/tutorials/planning_around_objects/planning_around_objects.html
- Movelt Task Constructor Pick & Place — https://moveit.picknik.ai/main/doc/tutorials/pick_and_place_with_moveit_task_constructor/pick_and_place_with_moveit_task_constructor.html
- Movelt Servo — https://moveit.picknik.ai/main/doc/examples/realtime_servo/realtime_servo_tutorial.html
- ros2_control JointTrajectoryController (Kilted) — https://control.ros.org/kilted/doc/ros2_controllers/joint_trajectory_controller/doc/userdoc.html
- ros2_control Force Torque Sensor Broadcaster (Kilted) — https://control.ros.org/kilted/doc/ros2_controllers/force_torque_sensor_broadcaster/doc/userdoc.html

Ja Movelt/ros2_control release līnijas vai API mainās, references workspace nedrīkst klusi pārslēgt uz citu distro. Vispirms jāiziet M25/M30 compatibility gates un jāatjauno versijas pierādījums.

68. References platformas v6.0 statuss

Rokasgrāmata v6.0 aptver M0–M30. M0–M24 izveido un validē mobilo autonomo platformu; M25–M30 pievieno manipulatora modelēšanu, Movelt planning, MTC pick/place, force/torque robežu, AI-assisted grasp kandidātus un pilnu mobile-manipulator capstone.

Bloks	Rezultāts
M0–M6	droša zemā līmeņa kustība, mērījumi, PID, odometrija, IMU, watchdog
M7–M12	ROS 2, EKF, LiDAR, SLAM, Nav2, Machine Vision

Bloks	Rezultāts
M13–M18	evidence, Digital Twin, fault injection, safety diagnostics, fusion, V&V
M19–M24	real-time boundary, CAN, CI/Docker, 3D, fleet, industrial release
M25–M30	mobile manipulation: MoveIt/MTC/force/AI/integrēta misija

Automātiskie host testi un simulācijas faili neaizstāj fizisku M0–M30 validāciju uz konkrēta robota. Fiziskās references rokas mehānika un drošība jāprojektē pēc konkrētā hardware datiem; EduArm6 kursa modelis ir programmatūras/kinemātikas mācību etalons.

69. M31–M36 — autonomās misijas inženierijas bloks

M31–M36 noslēdz vertikālo 2WD + mobile manipulator trasi ar sistēmas līmeņa autonomiju. Šajā posmā vairs nav pietiekami demonstrēt, ka atsevišķs mezgls “strādā”. Studentam jāparāda, ka pilna misija ir deterministiski vadāma, atkopjama, konservatīva pret nederīgiem datiem un pierādāma ar requirements → test → evidence ķēdi.

MISSION INTENT -> BT / EXECUTIVE -> FUSION -> SEMANTIC CONTEXT

-> OPTIMIZER CANDIDATE -> HARD GATES -> EXECUTION -> HEALTH -> EVIDENCE

AI, semantiskais slānis un optimizeris dod kandidātus vai kontekstu. Tie nedrīkst apiet collision checking, joint limits, base lock, sensor freshness, watchdog vai citus hard gates.

Milestone	Kodols	Obligātais acceptance pierādījums
M31	Mission Executive / Behavior Trees	bounded recovery + hard safety HOLD
M32	Advanced sensor fusion	residual/covariance/NIS + stale-data rejection
M33	Semantic mapping	zone/action permission + UNKNOWN_CONTEXT reject
M34	Trajectory/grasp optimization	hard-gated top-N candidate comparison
M35	Reliability & maintenance	health trend + critical HOLD + availability
M36	Autonomous mission capstone	3 nominal + recovery + safety HOLD + traceability

70. M31 — Mission Executive un Behavior Trees

M31 mērķis ir pāriet no vienas lielas lineāras state machine uz modulāru misijas loģiku. References modeli students strādā ar Sequence, Fallback, Retry un Safety Gate. Nav2 Behavior Tree Navigator kalpo kā industriāli reālistisks arhitektūras piemērs, bet kursa host-side mini BT ir tikai testējams mācību kodols.

70.1. Galvenais princips — bounded recovery

Recoverable kļūda drīkst tikt atkārtota tikai iepriekš definētu reižu skaitu. Bezgalīgs retry ir defekts, jo sistēma var palikt kustībā vai bloķēt resursus bez skaidra gala stāvokļa.

SUCCESS / FAILURE / RUNNING / HOLD

HOLD ir atsevišķs stāvoklis hard safety problēmām. Safety fault nedrīkst tikt maskēts kā parasts FAILURE, kuru BT automātiski atkārtos.

70.2. M31 acceptance

- Nominal tree pabeidz misiju ar SUCCESS.
- Viena apzināta navigation failure tiek atkopta bounded retry robežās.
- Pārsniegts retry limits dod FAILURE, nevis bezgalīgu ciklu.
- Hard safety fault dod HOLD bez parastā retry.
- Saglabāts tick/event log ar expected/actual rezultātu.

71. M32 — Advanced Sensor Fusion un Consistency Gating

M32 māca, ka sensor fusion nav vienkāršs vidējais. Students analizē residual (innovation), innovation covariance un Normalized Innovation Squared (NIS). Mērījums vispirms iziet freshness un consistency gate; tikai pēc tam to drīkst izmantot fused state novērtējumā.

$$\text{NIS} = \text{residual}^T * S^{-1} * \text{residual}$$

Kursa references host-side piemērā izmanto diagonālu innovation covariance, lai studentam būtu skaidri redzams katra residual ieguldījums. Chi-square robeža tiek konfigurēta atbilstoši mērījuma dimensijai un izvēlētajam confidence līmenim; tā nav “maģiska konstante”.

71.1. Obligātie negatīvie testi

- Outlier ar lielu residual tiek noraidīts kā INCONSISTENT.
- Stale timestamp tiek noraidīts neatkarīgi no residual lieluma.
- Nederīga vai nulles covariance izraisa konfigurācijas/testa kļūdu, nevis klusu fusion.
- Evidence satur residual, variance/covariance, NIS, threshold un acceptance decision.

72. M33 — Semantic Mapping un task context

Ģeometriskā karte atbild uz jautājumu “kur robots var pārvietoties?”, bet semantiskā karte — “ko konkrētajā vietā drīkst darīt?”. References mapē ir PICK_STATION, DROP_STATION, SLOW_ZONE un NO_GO zonas.

72.1. Conservative default

Ja koordinātei nav zināms semantiskais konteksts, mission executive nedrīkst pieņemt, ka task action ir atļauts. UNKNOWN_CONTEXT references sistēmā tiek interpretēts konservatīvi — action tiek noraidīts līdz validētam kontekstam.

72.2. M33 acceptance

- PICK ir atļauts tikai validētā pick zonā.
- PLACE ir atļauts tikai validētā drop zonā.
- NO_GO zona bloķē navigācijas/task darbību.
- SLOW_ZONE samazina speed limit atbilstoši konfigurācijai.
- Pārklājošu zonu gadījumā izmanto konservatīvāko attiecīgo limitu.

73. M34 — Trajectory un grasp multi-objective optimization

M34 pāriet no “pirmais derīgais plāns” uz vairāku derīgu kandidātu salīdzināšanu. Vispirms darbojas hard gates: collision-free un within-limits. Tikai pēc tam kandidāti tiek salīdzināti ar dokumentētu cost funkciju.

$$\begin{aligned} \text{cost} = & w1 * \text{path} + w2 * \text{time} + w3 * \text{energy} + w4 * (1 - \text{clearance}) \\ & + w5 * (1 - \text{joint_margin}) + w6 * (1 - \text{grasp_score}) \end{aligned}$$

Konkrētie svaru koeficienti ir kursa eksperimenta parametri, nevis universāli “pareizie” svāri. Studentam jāpierāda, kā svaru maiņa ietekmē izvēli un kāpēc izvēlētais kandidāts ir tehniski pamatots.

73.1. Hard gate pirms score

- Colliding kandidāts netiek “izglābts” ar labu AI grasp score.
- Joint-limit pārkāpums netiek kompensēts ar īsu trajektoriju.
- Top-N rezultātos saglabā cost term sadalījumu, ne tikai gala score.

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

74. M35 — Reliability, diagnostics un maintenance analytics

M35 savieno ROS diagnostics domāšanu ar vienkāršu reliability analytics. References health score izmanto battery, CPU temperatūru, encoder drop, LiDAR invalid ratio un motor current trendus. Tomēr critical freshness/watchdog problēma ir hard HOLD un nedrīkst pazust aggregate score vidējā.

74.1. Health states

OK -> WARN -> MAINTENANCE -> HOLD

Sliekšņi references paketē ir mācību sākuma punkti. Fiziskam robotam tie jākalibrē ar konkrēto sensoru, bateriju, motoru, datoru un lietošanas vidi.

74.2. Reliability pamatmetrikas

Availability = MTBF / (MTBF + MTTR)

MTBF un MTTR aprēķinam nepieciešami reāli failure/repair pieraksti. Viena laboratorijas sesija nav pietiekama, lai apgalvotu produkta uzticamību; mērķis ir iemācīt datu un lēmuma ķēdi.

75. M36 — Autonomous Mission Capstone un System V&V

M36 apvieno visu M0–M35 vienā controlled mission. References nominal scenārijs: preflight → navigate to pick → semantic context → fusion consistency → candidate optimization → pick → grasp verify → stow → deliver → place → evidence.

75.1. Obligātie scenāriji

Scenārijs	Expected result
3 × nominal mission	SUCCESS bez neparedzētas bāzes/rokas kustības
Recoverable navigation failure	Recovery darbojas bounded retry robežās un misija pabeidzas
Fusion outlier vai stale sensor	Mērījums tiek noraidīts; task neturpina uz nederīga data pamata
Semantic UNKNOWN / NO_GO	Action tiek konservatīvi bloķēts
Hard safety / watchdog fault	HOLD; nav automātiska resume

75.2. Gala evidence pack

- software/config version un git commit/tag;
- calibration parametri;
- requirements → test → evidence matrica;
- nominal un fault rosbag/log/CSV;
- known limitations un unresolved risks;
- reviewer RELEASE/HOLD lēmums.

76. M31–M36 pasniedzēja sagatavošanas checklist

Pirms laboratorijas	[]
References Kilted/Nav2/MoveIt vide nofiksēta	[]
M0–M30 ieejas gate statusi pārbaudīti	[]
Fault injection zona un fiziskais stop pieejams	[]
Semantic map un mission scenāriji sagatavoti	[]
Acceptance workbook izdalīts	[]

Pirms laboratorijas	[]
Evidence mapju nosaukumi / repo branch sagatavoti	[]
Studentiem izskaidrota atšķirība starp software HOLD un fizisku E-stop	[]

77. M31–M36 troubleshooting karte

Simptoms	Vispirms pārbaudi	Nākamais solis
BT iestrēgst	bounded retry un node status	event log / blackboard inputs
Fusion noraida visu	units, covariance, timestamp	residual/NIS decomposition
Semantic action vienmēr rejected	map frame un zone bounds	requested action permissions
Optimizer izvēlas dīvainu kandidātu	hard gates un normalization	weights + cost terms
Health score lēkā	sample period / sensor quality	EWMA alpha un raw metrics
M36 nominal misija nestabila	izolē pēdējo stabilo milestone	rosbag replay + fault localization

78. M31–M36 tehniskie avoti un references statuss

Pārbaudīts 2026-08-19. Kursa references izmanto oficiālo Nav2, MoveIt un ROS 2 diagnostics dokumentāciju. Semantic map, health score un host-side BT/fusion kodols ir kursa mācību implementācijas.

- Nav2 Behavior Trees — https://docs.nav2.org/behavior_trees/
- Nav2 BT Navigator — <https://docs.nav2.org/configuration/packages/configuring-bt-navigator.html>
- Nav2 custom BT plugin tutorial — https://docs.nav2.org/plugin_tutorials/docs/writing_new_bt_plugin.html
- Nav2 Kilted migration / SubTrees — <https://docs.nav2.org/migration/Kilted.html>
- MoveIt planning adapters — https://moveit.picknik.ai/main/doc/examples/planning_adapters/planning_adapters_tutorial.html
- MoveIt parallel planning — https://moveit.picknik.ai/main/doc/how_to_guides/parallel_planning/parallel_planning_tutorial.html
- ROS 2 diagnostic_updater Kilted — https://docs.ros.org/en/ros2_packages/kilted/api/diagnostic_updater/

79. References platformas v7.0 statuss

Rokasgrāmata v7.0 aptver M0–M36. M0–M12 izveido autonomu mobilo bāzi, M13–M24 pievieno validāciju un industrijas paplašinājumus, M25–M30 — mobile manipulatoru, bet M31–M36 noslēdz ar mission executive, fusion consistency, semantic context, optimization, reliability analytics un pilnu autonomous mission V&V.

Automātiskie host testi un simulācijas faili neaizstāj fizisku M0–M36 validāciju. Gala RELEASE šajā rokasgrāmatā ir kursa acceptance statuss, nevis produkta funkcionālās drošības vai normatīvās atbilstības sertifikācija.

80. M37–M42 — pētniecības robotikas bloks

M37–M42 turpina references 2WD + mobile manipulator platformu pēc M36. Šī bloka mērķis nav vienkārši pievienot vēl vienu sensoru vai AI modeli, bet iemācīt pētniecības eksperimenta disciplīnu: reproducējamu vidi, baseline, negatīvos testus, deterministiskus drošības gates, metriku definīciju un pierādāmu gala benchmarku.

SIM / DATA / HIGH-LEVEL MODEL -> CANDIDATE -> HARD GATES -> MISSION EXECUTIVE -> ROBOT -> EVIDENCE

Learned, semantic vai language/VLM komponents references arhitektūrā nekad nav tiešs aktuatora komandētājs. Tas ģenerē kandidātu, kuru pirms izpildes pārbauda state, freshness, semantic permissions, collision/limits, mission mode un watchdog slāņi.

Gate	Tēma	Obligātais pierādījums
M37	Reinforcement Learning simulatorā	fixed-seed eval + baseline + safety shield
M38	Active Perception / Next-Best-View	feasible view + izmērāms observation improvement
M39	Semantic/VLM task planning	strukturēts skill plan + schema/context rejection
M40	Model Predictive Control	prediction horizon + constraints + fallback
M41	Cybersecurity / defensive red-team	least privilege + deny + safe state
M42	Research benchmark	provenance + repeated scenarios + traceability

81. M37 — Reinforcement Learning tikai simulatorā

M37 sākas ar custom environment un skaidru state/action/reward/termination formulējumu. References paketē ir dependency-light GridWorld tipa vide un Q-learning baseline, lai students var saprast eksperimenta principu arī bez smagas ML infrastruktūras. Pēc tam advanced ceļš var izmantot Gymnasium un PPO/Stable-Baselines3.

81.1. Safety shield

Policy piedāvā action, bet environment/safety shield pārbauda robežas un šķēršļus. Nederīgs action netiek izpildīts. Fiziska policy deployment nav M37 PASS prasība, un rezultātu nedrīkst interpretēt kā pierādījumu, ka policy ir droša uz īsta robota.

81.2. Reproducibility gate

- Fiksēti train un evaluation seeds.
- Training/tuning scenāriji nodalīti no gala evaluation.
- Saglabāts random vai cits non-learning baseline.
- Ziņots success rate, mean steps un shield interventions.
- Vismaz viens reward/parameter sensitivity eksperiments.

82. M38 — Active Perception un Next-Best-View

M38 māca robotam ne tikai pasīvi pieņemt sensoru informāciju, bet izvēlēties nākamo skata punktu, ja pašreizējā uztvere ir nepietiekama. References modelī katrs candidate viewpoint vispirms iziet feasibility gate; tikai pēc tam tas tiek vērtēts pēc visibility, uncertainty reduction un motion cost.

$$\text{score} = w_{\text{vis}} * \text{visibility} + w_{\text{info}} * \text{uncertainty_reduction} - w_{\text{cost}} * \text{travel_cost}$$

Skaitliskie svāri ir eksperimenta parametri, nevis universālas konstantes. Studentam jāpierāda, ka selected view ir sasniedzams un ka pēc kustības uztveres kvalitāte uzlabojas vai sistēma korekti secina, ka uzlabojams skats nav pieejams.

82.1. Negatīvie testi

- Labākais score, bet unreachable candidate — obligāti jānoraida.
- Candidate atrodas semantiski aizliegtā zonā — jānoraida neatkarīgi no score.
- Visi candidate ir infeasible — rezultāts ir NO_VALID_VIEW/HOLD, nevis patvaļīga kustība.

83. M39 — Semantic/VLM-assisted task planning

M39 ievieš provider-agnostic augsta līmeņa task planning robežu. Language/VLM modulis drīkst ierosināt tikai strukturētu skill sarakstu, piemēram NAVIGATE, OBSERVE, PICK, PLACE, WAIT vai DOCK. Nevienas brīva teksta fragments netiek interpretēts kā tieša robota komanda.

INTENT -> structured skill plan -> schema validator -> semantic/context gate -> M31 mission executive

83.1. Obligātie rejection gates

- Nezināms skill, piemēram RAW_MOTOR.
- NAVIGATE/PICK/PLACE mērķis atrodas NO_GO zonā.
- PLACE bez iepriekšēja PICK.
- Pārāk gara vai cikliska darbību secība.
- Plāns mēģina apiet mission executive vai safety gate.

Šī arhitektūra ir saderīga ar pētījumu virzienu, kur valodas modelis dod augsta līmeņa semantiku, bet reālās robota prasmes un to feasibility ierobežo izpildi.

84. M40 — Model Predictive Control

M40 demonstrē MPC būtību: sistēmas modelis paredz nākotnes stāvokļus vairākus soļus uz priekšu, kandidāti tiek novērtēti ar cost function un constraints, bet izpilda tikai pirmo izvēlēto control action; nākamajā ciklā optimizāciju atkārtoti ar jaunāko stāvokli.

model -> prediction horizon -> cost + constraints -> u0 -> measure -> repeat

84.1. References un advanced trase

Pakotnē iekļautais finite-horizon kontrolieris ir dependency-light mācību baseline, nevis pilnvērtīgs NMPC solveris. Advanced darbā studenti var portēt to uz acados closed-loop examples un salīdzināt solve time, tracking un constraint behavior. acados ir paredzēts ātrai nonlinear optimal control/NMPC un MHE risināšanai, tostarp real-time/embedded lietojumiem.

84.2. M40 PASS

- v un ω limits nekad netiek pārsniegti;
- šķēršļu/clearance ierobežojums ir testēts;
- ir baseline salīdzinājums;
- mērīts compute/solve time;
- solver/no-feasible-command gadījumam ir STOP/HOLD fallback.

85. M41 — Cybersecurity un defensive red-team validācija

M41 ir aizsardzības laboratorija. Tā neapmāca uzlauzt robotu. Students definē assets, trust boundaries, least-privilege publish/subscribe nepieciešamību un pārbauda, ka neatļauts komunikācijas pieprasījums tiek noraidīts, neizraisot kustību.

85.1. References princips

IDENTITY / ENCLAVE -> LEAST PRIVILEGE POLICY -> DENY OUTSIDE POLICY -> SAFE STATE

ROS 2 Kilted references vidē advanced daļa balstās uz SROS2 keystore, enclaves un permission/policy rīkiem. Privātas atslēgas un keystore artefakti nedrīkst tikt commitoti kursa repozitorijā.

85.2. Obligātie drošie negatīvie testi

- Perception node mēģina publicēt /cmd_vel — DENY.
- Node mēģina izmantot neparedzētu topic — DENY.
- Trūkst vai ir nederīga security konfigurācija — sistēma neiet automātiskā kustībā.
- Repository scan atrod privātas atslēgas failu — release HOLD.

86. M42 — Research Benchmark un reproducibility

M42 pārvērš visu M0–M41 platformu par atkārtojamu eksperimentu. Viena veiksmīga demonstrācija nav benchmark. Katram run jābūt scenario ID, seed/config, software/hardware context, raw evidence un iepriekš definētām metrikām.

86.1. Obligātā benchmark matrica

- vismaz 3 nominal runs;
- vismaz 1 recoverable failure;
- vismaz 1 hard safety HOLD;
- vismaz 1 semantic/VLM rejection;
- vismaz 1 perception/sensor degradation;
- requirement -> test -> evidence traceability.

86.2. Metrikas

success rate | safety violations | recovery count | task quality | latency | steps/path | resource usage

RobotPerf un ROS benchmarking metodikas tiek izmantotas kā inspirācija reproducējamai, kvantitatīvai veikspējas mērīšanai. REP-2014 dokumenta statuss ir Rejected, tāpēc tas rokasgrāmatā tiek izmantots tikai kā metodoloģisks fons, nevis kā spēkā esošs ROS standarts.

87. M37–M42 pasniedzēja un studenta gala checklist

Punkts	Ko pārbaudīt	Statuss
Research question	Ir precīzi definēts, ko eksperiments mēra un ar ko salīdzina.	☐ PASS ☐ FAIL
Baseline	Ir vismaz viens saprotams non-learning/non-advanced baseline.	☐ PASS ☐ FAIL
Seeds/config	Seeds un konfigurācija ir saglabāti un atkārtojami.	☐ PASS ☐ FAIL

Punkts	Ko pārbaudīt	Statuss
Hard gates	Learned/VLM/optimizer output nevar apiet safety/limits/context gates.	☐ PASS ☐ FAIL
Negative tests	Ir iepriekš definēti rejection, fault un HOLD scenāriji.	☐ PASS ☐ FAIL
Evidence	Raw log/CSV/rosbag2 un summary metrics ir saglabāti.	☐ PASS ☐ FAIL
No cherry-pick	Ziņoti arī neveiksmīgi run, ne tikai labākais demo.	☐ PASS ☐ FAIL
Release	Synthetic PASS nav nosaukts par fizisku validāciju.	☐ PASS ☐ FAIL

88. M37–M42 tehniskie avoti un references statuss

Pārbaudīts 2026-08-19. Avoti ir saglabāti arī projekta `research_extension/SOURCES.md`. Tie ir references materiāli; konkrēta laboratorijas vide pirms kursa atkārtotas palaišanas jāpārbauda pret aktuālo programmatūras versiju.

- Gymnasium custom environment — https://gymnasium.farama.org/main/introduction/create_custom_env/
- Stable-Baselines3 PPO — <https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html>
- OA-NBV active perception paper — <https://arxiv.org/abs/2603.11072>
- SayCan / grounded language planning — <https://arxiv.org/abs/2204.01691>
- acados documentation — <https://docs.acados.org/>
- ROS 2 Kilted SROS2 — <https://docs.ros.org/en/kilted/p/sros2/>
- RobotPerf benchmark suite — <https://github.com/robotperf/benchmarks>
- REP-2014 benchmarking performance — <https://ros.org/reps/rep-2014.html> (Status: Rejected; metodoloģisks fons tikai)

89. References platformas v8.0 statuss

Rokasgrāmata v8.0 aptver M0–M42. M0–M12 izveido autonomu 2WD bāzi; M13–M24 pievieno validāciju, industriālu integrāciju un release disciplīnu; M25–M30 pievieno mobile manipulatoru; M31–M36 sistēmas līmeņa autonomiju; M37–M42 — reproducējamu pētniecības robotiku ar RL simulatorā, active perception, strukturētu semantic/VLM task planning, MPC, defensive cybersecurity un gala benchmarku.

Automātiskie host-side testi pārbauda references algoritmu un failu kontraktus, bet neaizstāj reālu ROS 2/MoveIt/SROS2/Gazebo aparātūras vai tikla validāciju. Fiziskais M0–M42 RELEASE ir iegūstams tikai pēc faktiskajiem acceptance testiem uz konkrētās platformas.

90. M43–M48 — thesis-grade / publication-grade pētījuma bloks

M43–M48 pārvērš M0–M42 sistēmu no demonstrējamās robotikas platformas par auditējamu pētījuma vidi. Galvenais produkts vairs nav tikai robots, bet pierādījumu ķēde, kuru cits students vai recenzents var atkārtot un pārbaudīt.

DATASET -> HASH/PROVENANCE -> RUN ID/CONFIG/SEED -> RAW EVIDENCE -> TRACE -> STATISTICS -> ENERGY -> THESIS PACK -> RELEASE/HOLD

Obligātais princips: synthetic smoke tests, simulācija un fiziskā robota rezultāti tiek marķēti atsevišķi. Trūkstošs raw evidence netiek aizstāts ar manuāli ievadītu "PASS".

91. M43 — Dataset governance un provenance

Pētījumā dataset ir konfigurācijas sastāvdaļa, nevis nejauša mape ar failiem. Katram gala rezultātam jābūt sasaistītam ar konkrētu dataset ID un SHA-256 identitāti.

Dataset card dokumentē mērķi, sastāvu, kolekcijas metodi, sensorus, vienības, split stratēģiju, zināmās nepilnības, licencēšanas/piekļuves nosacījumus un out-of-scope lietojumus.

dataset_id + version + manifest_sha256 + card + provenance + split/leakage review

- file-level SHA-256 + whole-dataset hash;
- schema un vienību dokumentācija;
- train/validation/test vai calibration/evaluation leakage pārbaude;
- fizisks/simulēts/mixed status;
- failed/izslēgto datu iemesli;
- sensitive/personal data status un piekļuves ierobežojumi.

92. M44 — Experiment tracking un konfigurācijas identitāte

M44 ievieš run kā pirmās klases artefaktu. "Eksperiments ar PID 2" nav pietiekams nosaukums; nepieciešams unikāls run ID, dataset ID, config hash, seed, code commit un raw artifact ceļš.

Dependency-free local tracker ir obligātais minimums; MLflow ir optional profesionālais ceļš eksperimentu, parametru, metriku un artefaktu centralizētai pārvaldībai.

run_id -> dataset_id | config_sha256 | seed | git_commit | scenario | metrics | artifacts

- baseline un variants tiek palaisti ar dokumentētu atšķirību;
- seed un scenario ID saglabājas;
- failed runs netiek dzēsti;
- config drift tiek atklāts ar hash;
- raw rosbag/log/trace artefakti tiek piesaistīti run ID.

93. M45 — ROS 2 tracing un profiling

M45 mēra izpildlaika uzvedību ar sadalījumu, nevis tikai vidējo. ROS 2 Lyrical oficiālais tracing ceļš izmanto ros2_tracing/ros2trace un tracertools_analysis.

P99, max un deadline misses ir īpaši svarīgi robotikā, jo viena reta aizkave var būt nozīmīgāka par labu vidējo rezultātu.

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

trace -> callback/stage durations -> p50/p95/p99/max -> deadline misses -> evidence

- p50, p95, p99, max callback/stage duration;
- deadline miss count un ratio;
- scenario/run/config sasaistīšana;
- trace path saglabāšana;
- outlier analīze un cēloņa hipotēze.

94. M46 — Statistical comparison un ablation studies

M46 aizliedz secinājumu “variants B ir labāks” no viena run. Baseline un variants tiek atkārtoti kontrolētā dizainā, un tiek ziņota uncertainty un effect size.

P-value nav vienīgais pierādījums. Gala secinājumā jāparāda arī efekta lielums, confidence interval un praktiskā inženiertehniskā nozīme.

baseline -> repeated runs -> variant -> CI + effect size + test -> ablation -> conclusion

- atkārtojumu skaits ir deklarēts;
- paired/independent dizains ir pamatots;
- 95% confidence interval;
- effect size;
- failed runs/exclusions ir redzami;
- ablation matrica izolē vienas komponentes ietekmi.

95. M47 — Energy / performance benchmarking

M47 pievieno enerģiju kā sistēmas metriku. Linux powercap/RAPL var sniegt hosta compute enerģijas skaitītāju, bet tas nav automātiski visa robota enerģijas patēriņš.

Fiziskam robotam ieteicams ārējs power logger pie skaidri definētas elektriskās robežas. Salīdzinājumam jāizmanto vienāds workload, warm-up un battery/host state.

efficiency = successful tasks / energy [J]

- measurement boundary ir skaidri nosaukta;
- energy [J], duration [s], average power [W];
- success/task quality tiek mērīts tajā pašā run;
- workload un sensoru frekvences ir fiksētas;
- counter wrap vai mēraparāta kalibrācija ir dokumentēta.

96. M48 — Thesis evidence pack un reproducibility release

M48 automātiski savāc M43–M47 artefaktus, ģenerē provenance un SHA-256 manifestu, un izdod RELEASE/HOLD. Ja mandatory pierādījums nav atrodams, rezultāts ir HOLD.

Šis ir gala tilts uz diplomdarbu, bakalaura darbu, conference demo vai publication supplement.

mandatory evidence complete ? RELEASE : HOLD

- research question/hypothesis;
- dataset card + manifest;
- run registry/config/seeds;

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

- raw logs/rosbag2/traces;
- statistika/ablations;
- energy/performance data;
- failed-run/exclusions log;
- limitations;
- clean rerun record;
- requirement-test-evidence matrix;
- final SHA-256 manifest un reviewer decision.

97. M43–M48 pasniedzēja/studenta acceptance checklist

Gate	Studentam jāpierāda	Pasniedzēja lēmums
M43	Dataset card, manifest/hash, provenance, leakage review	PASS/FAIL
M44	Run ID, dataset/config/seed/code, artifacts	PASS/FAIL
M45	Trace path, p50/p95/p99/max, deadline misses	PASS/FAIL
M46	Repeated comparison, CI, effect size, ablation	PASS/FAIL
M47	Energy boundary, workload, energy + task quality	PASS/FAIL
M48	No missing mandatory evidence; clean rerun; final manifest	RELEASE/HOLD

98. M43–M48 tehniskie avoti un references statuss

Pārbaudīts 2026-08-19. Rīku versijas ir jāfiksē konkrētajā eksperimentā; šie URL ir metodoloģiskās references.

- ROS 2 Lyrical tracing — <https://docs.ros.org/en/lyrical/Tutorials/Advanced/ROS2-Tracing-Trace-and-Analyze.html>
- MLflow Tracking — <https://mlflow.org/docs/latest/tracking>
- DVC — <https://dvc.org/doc>
- FAIR Principles — <https://www.go-fair.org/fair-principles/>
- Datasheets for Datasets — <https://arxiv.org/abs/1803.09010>
- SciPy bootstrap — <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.bootstrap.html>
- SciPy Mann-Whitney U — <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html>
- Linux powercap/RAPL — <https://www.kernel.org/doc/html/latest/power/powercap/powercap.html>

99. References platformas v9.0 statuss

Rokasgrāmata v9.0 aptver M0–M48. M43–M48 nepievieno jaunu kustības funkciju robotam; tas pievieno dataset, experiment, timing, statistics, energy un thesis-release disciplīnu, lai M0–M42 rezultātus var izmantot akadēmiski aizstāvamā pētījumā.

Host-side testi pārbauda references rīku loģiku, bet fiziskais M0–M48 RELEASE ir iespējams tikai pēc reālās aparātūras/ROS 2/MoveIt/trace/energy eksperimentu izpildes un pierādījumu pārskatīšanas.

Fiziskā 2WD mācību robota uzbūves rokasgrāmata v11.0 | Programmēšanas un datorizētu sistēmu pamati

M55–M60 — publikācija, reproducējamība un tehnoloģijas nodošana

Šis bloks noslēdz M0–M60 inženierijas trasi. M55–M60 mērķis nav pievienot robotam jaunu funkciju, bet pierādīt, ka rezultātus var publicēt, atkārtot, kontrolēti izlaist, demonstrēt atbilstošā vidē un nodot citam uzturētājam bez slēptām zināšanām.

M55 — Manuscript / claim-to-evidence

- Katrs būtiskais apgalvojums tiek sasaistīts ar evidence ID, grafiku/tabulu un ierobežojumu.
- READY tikai tad, ja kvantitatīvam apgalvojumam ir reproducējams avots; citādi HOLD.
- Praktiskais rezultāts: manuscript claim matrix un reviewer checklist.

Expected	Actual	Evidence	PASS/FAIL

M56 — Independent reproducibility audit

- Fresh checkout vai release archive tiek uzstādīts tīrā vidē, neizmantojot autora lokālos failus.
- Pārbauda dataset/config hash, testus un vienu reprezentatīvu benchmarku ar iepriekš definētu toleranci.
- Visi workaround un neatbilstības tiek pierakstītas; reproducibility audit nav “palīdzēt līdz strādā”.

Expected	Actual	Evidence	PASS/FAIL

M57 — Research software release

- Release satur skaidru versiju, LICENSE ar korektu SPDX identifikatoru, CITATION.cff, README, CHANGELOG, archive SHA-256 un known limitations.
- GitHub atbalsta CITATION.cff, lai programmatūrai varētu ģenerēt citēšanas informāciju; publisks repo bez licences nav automātiski atvērtais kods.
- Public release pirms secrets/private-data audit nav pieļaujams.

Expected	Actual	Evidence	PASS/FAIL

M58 — TRL un pilot readiness

- TRL tiek lietots kā tehnoloģijas brieduma references skala, nevis reklāmas skaitlis.
- Mācību robotam TRL apgalvojums tiek ierobežots ar faktisko pierādījumu vidi: laboratorija, relevant environment, prototype demonstration.
- Pilotam obligāti: entry criteria, abort criteria, acceptance criteria, supervision, rollback un īpašnieks.

Expected	Actual	Evidence	PASS/FAIL

M59 — Maintenance / technology transfer

- Katram uzturējamam elementam definē owner, intervālu, metodi, acceptance un record location.

- RACI, backup/restore, rollback, kalibrācijas, rezerves daļu un apmācības pierādījumi ir nodošanas sastāvdaļa.
- Transfer PASS tikai tad, ja sistēmu var uzturēt persona, kas nav sākotnējais izstrādātājs.

Expected	Actual	Evidence	PASS/FAIL

M60 — Lifecycle release / operational handover

- Final gate apvieno M55–M59 ar operational demo, support boundary un known limitations.
- Jebkurš obligāts NOT RUN/FAIL bloķē RELEASE un dod HOLD.
- Izglītības projekta RELEASE nenozīmē funkcionālās drošības sertifikāciju vai komerciālu atbilstības novērtējumu.

Expected	Actual	Evidence	PASS/FAIL

M0–M60 gala release princips

PROBLĒMA -> PRASĪBA -> FUNKCIONĀLAIS SADALĪJUMS -> IMPLEMENTĀCIJA -> TESTS -> EVIDENCE -> REPRODUCIBILITY -> RELEASE/HOLD -> UZTURĒŠANA.

Gala release drīkst būt tikai RELEASE vai HOLD. “Gandrīz gatavs” nav acceptance statuss.

M61–M66 — Productizācija, ražošana un tirgus izlaišanas simulācija

Šis bloks paplašina M0–M60 robotikas trasi līdz produkta izstrādes un tirgus izlaišanas mācību simulācijai. Tas nepadara prototipu automātiski CE atbilstošu.

M61 — Stakeholder / URS

Pārvērst stakeholder vajadzības pārbaudāmās lietotāja prasībās ar unikāliem URS ID, prioritāti, racionāli un acceptance criterion.

M62 — Product requirements

Sadalīt URS SYS/HW/SW/SAF/SEC/UI/MFG prasībās un izveidot divvirzienu traceability līdz testam un evidence.

M63 — FMEA / risk management

Identificēt hazard/failure mode, novērtēt S/O/D, definēt risk controls un pieņemt tikai dokumentētu residual risk.

M64 — DFM/DFA + production BOM

Pārveidot prototipa BOM par ražošanas BOM ar part number, alternatīvām, lead time, inspection un assembly/control plan.

M65 — Verification & Validation Master Plan

Nofiksēt verification metodes, testu līmeņus, konfigurāciju, preconditions, expected/actual/evidence un deviation kontroli.

M66 — CE / regulatory market-release simulation

Izveidot applicability matrix, technical-file indeksu un simulētu DoC/CE release gate. Šī ir mācību simulācija, nevis juridisks atbilstības novērtējums.

M0–M66 gala princips

No prasības līdz evidence: Stakeholder need -> URS -> Product requirement -> Risk control -> Design/BOM -> Verification test -> Evidence -> Release decision. Ja kādē trūkst obligāta posma, gala statuss ir HOLD.