

KMJ DYNAMICS

AUTO-CART

The Autonomous Beach Cart — Complete Build Guide. The real working manual, expensive lessons included



● RC build — build only the green

● Autonomy — the full follow-me stack

Edition 2026-08-14 r4 · kmjdynamics.com · hello@kmjdynamics.com
Free lifetime updates — this document improves as the cart does.

This is the free preview — every  block is real, tested content: the exact settings, part numbers, wiring rules, and the failures that produced them. Get the Complete Build Package at kmjdynamics.com

Contents

- 0. Overview & build order
- 1. Frame & chassis  [RC BUILD]
- 2. Steering  [RC BUILD]
- 3. Drivetrain (motors, chain, wheels)  [RC BUILD]
- 4. Power system & distribution  [RC BUILD — Buck #2 is  autonomy-only]
- 5. Compute (Raspberry Pi 5)  [AUTONOMY ONLY — the RC build has no Pi]
- 6. Motor controller (VESC Duet XS 60V dual VESC — VESC Labs)  [RC BUILD]
- 7. RC control  [RC BUILD — see the RC-only wiring box below]
- 8. Drive & steering software (drive.py / steering.py)  [AUTONOMY ONLY]
- 9. Sensors & telemetry  [AUTONOMY ONLY]
- 10. Displays (camera + 5" screen)  [AUTONOMY ONLY]
- 11. UWB localization — WORKING (4 anchors + tag, [4A] fix @3.3Hz)  [AUTONOMY ONLY]
- 12. Follow-me autonomy (modules/follow_me.py) — WORKING, road-tested  [AUTONOMY ONLY]
- 13. Safety systems (the stack, in order of what fires)  +  [VESC-side rows apply to RC too]
- 14. Networking, web panel & remote access  [AUTONOMY ONLY]
- 15. Software, running & autostart  [AUTONOMY ONLY]
- 16. 3D-printed parts (generators in 3d_models/, PETG unless noted)  +  [wheel hubs + battery box = RC too]
- 17. Key pin & config reference (full map: PINOUT.md)
- 18. Open items as of 8/3

Why this guide keeps getting better parts

I built this in layers on purpose. First a "dumb wagon" — a frame that rolls, steers, and drives by remote — and I made sure that worked before any self-driving stuff went on. Test each layer by itself before stacking the next one on top. That way, when something acts up, you know which layer to blame. I know the autonomous parts are the fun parts. But every hour I spent making a layer solid saved me five hours of head-scratching later. Trust me on this one.

Build the mechanical cart first (frame, steering, drivetrain), then power, then the computer and motor controller, then RC control, then sensors and displays, then the autonomy hardware (UWB) and the follow-me software, and finally the safety systems that wrap all of it.

0. Overview & build order

IN PLAIN ENGLISH

I built this in layers on purpose. First a "dumb wagon" — a frame that rolls, steers, and drives by remote — and I made sure that worked before any self-driving stuff went on. Test each layer by itself before stacking the next one on top. That way, when something acts up, you know which layer to blame. I know the autonomous parts are the fun parts. But every hour I spent making a layer solid saved me five hours of head-scratching later. Trust me on this one.

Build the mechanical cart first (frame, steering, drivetrain), then power, then the computer and motor controller, then RC control, then sensors and displays, then the autonomy hardware (UWB) and the follow-me software, and finally the safety systems that wrap all of it.

Cost and scope, up front: budget **\$3,500–4,100 in parts for the RC build**, or **\$4,300–5,100 for the full autonomous build** — the ● autonomy stack is roughly \$800–1,000 on top of the ● one. Where you land inside those ranges depends mostly on what your shop already stocks; STOCK_LIST.md breaks it down and names the two lines you can't shop your way out of. On top of the parts you need a 3D printer that can run carbon-fiber nylon and the ability to weld aluminum — if you have neither, that's another \$600–1,500 before you cut anything. **The RC build is the cheaper and simpler path** — build only the ● chapters (§1–4, §6–7, plus the VESC-side rows of §13) and skip the entire lidar/UWB/sensor/compute stack; that's not how this cart was built, but those chapters stand alone, and the RC-only wiring box in §7 shows the simpler no-Pi hookup. **Two areas deliberately need your own ingenuity:** fitting the electronics into a weatherproof enclosure, and mounting everything to the frame to survive sand, salt, and vibration. The layout shown here is ONE working answer, not the only one — your parts and cable runs will differ.

● / ● **COLOR KEY (added 2026-08-09, launch prep):** every chapter here — and every section and line of STOCK_LIST.md — is tagged for which build actually needs it, because most builders only want the RC cart and should not overbuild for autonomy they aren't adding:

-  = **RC BUILD**. Required for the remote-control-only cart (the autonomous build uses all of this too — it's a superset). **Building RC? Buy and build ONLY the green items, then stop.**
-  = **AUTONOMY ONLY**. Only for the follow-me build. RC builders skip these chapters and parts entirely — this is the whole Pi/lidar/UWB/sensor stack, roughly \$800–1,000 of parts.
-  +  = **mixed** — mostly one, with the exceptions flagged inside the chapter.

(In the sellable PDF/HTML edition,  content renders with a green highlight and  with a purple one; in this Markdown master the dots carry the color.)

Since r3, STOCK_LIST.md additionally tags a handful of shop-consumable rows "**(OPTIONAL — shop stock)**" — wire, connector kits, heat shrink and the like that most shops already have. Use what you have; buy only what's missing. They stay  — the tier is about whether to BUY, not whether the build needs the function.

- 1.  Frame / chassis — 1.5" square aluminum, TIG welded, removable 1-1/4" side rails
 - 1.  Steering — wagon steering bracket + tie rods + 24V servo
 - 1.  Drivetrain — 2 BLDC motors, 5:1 chain, 16.5" wheels (BOND THE TIRES FIRST)
 - 1.  Power system — 24V 50Ah LiFePO4 (BMS-direct gauge), fuse, TVS, 5V buck, star ground (Buck #2 for the UWB bus = )
 - 1.  Compute — Raspberry Pi 5 + Argon NEO 5 + OS + cooling (RC build: no Pi at all)
 - 1.  Motor controller — VESC Duet XS 60V dual VESC (VESC Labs), configured over VESC Tool + CAN
 - 1.  RC control — DumboRC X4/X6F radio (RC-only builds wire it straight to the VESC/servo — see §7)
 - 1.  Drive & steering software — hardware PWM, ramps, auto park-brake, deadman
 - 1.  Sensors & telemetry — 6x TFmini ring, RPLIDAR S3, BMS-direct battery gauge over CAN/BLE
 - 1.  Displays — camera to the phone, radar on the 5" screen
 - 1.  UWB localization — 4 corner anchors + wearable tag (WORKING, 3.3Hz 4-anchor fix)
 - 1.  Follow-me autonomy — 4-corner solver + bearing PD steering (road-tested)
 - 1.  +  Safety systems — VESC-side failsafes are  for every build; the sensor gate stack is 
 - 1.  Networking, web panel & remote access
 - 1.  Software, autostart & backup
 - 1.  +  3D-printed parts — wheel hubs + battery box ; sensor mounts, screen case, tag case 
 - 1. Pin & config reference (autonomous cart)
 - 1. Open items as of 7/11
-

1. Frame & chassis ● [RC BUILD]

IN PLAIN ENGLISH

The frame is a flat rectangle of hollow square aluminum tube — picture a bed frame — 32 inches wide and 4 feet long, with extra tubes across it wherever something heavy sits (motors, axle, battery). I went with aluminum because it's light and won't rust at the beach, and welded because bolted joints shake loose in sand and vibration. Everything else in this guide bolts to this rectangle, so take your time getting it *square and flat*. A crooked frame means crooked wheels and a chain that keeps jumping off.

Materials —

1. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
2. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
3. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
4. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')

5. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
6. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
7. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
8. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
9. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
10. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')

11. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
12. 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')

#				
1	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	\$10.00
2	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") (32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") + 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125"))	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") (32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") + 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125"))	\$10.00 + 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")
+	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") (32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") + 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125"))	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	\$10.00 + 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")
1	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") (32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") + 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125"))	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")	\$10.00 + 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") 32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125")

32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')
32" x 48" x .125" aluminum tube (1/2" x 1/2" x .125") — 10' (10' x 10' x 10')

1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9
10	10	10	10	10	10
11	11	11	11	11	11
12	12	12	12	12	12
13	13	13	13	13	13
14	14	14	14	14	14
15	15	15	15	15	15
16	16	16	16	16	16
17	17	17	17	17	17
18	18	18	18	18	18
19	19	19	19	19	19
20	20	20	20	20	20
21	21	21	21	21	21
22	22	22	22	22	22
23	23	23	23	23	23
24	24	24	24	24	24
25	25	25	25	25	25
26	26	26	26	26	26
27	27	27	27	27	27
28	28	28	28	28	28
29	29	29	29	29	29
30	30	30	30	30	30
31	31	31	31	31	31
32	32	32	32	32	32
33	33	33	33	33	33
34	34	34	34	34	34
35	35	35	35	35	35
36	36	36	36	36	36
37	37	37	37	37	37
38	38	38	38	38	38
39	39	39	39	39	39
40	40	40	40	40	40
41	41	41	41	41	41
42	42	42	42	42	42
43	43	43	43	43	43
44	44	44	44	44	44
45	45	45	45	45	45
46	46	46	46	46	46
47	47	47	47	47	47
48	48	48	48	48	48
49	49	49	49	49	49
50	50	50	50	50	50
51	51	51	51	51	51
52	52	52	52	52	52
53	53	53	53	53	53
54	54	54	54	54	54
55	55	55	55	55	55
56	56	56	56	56	56
57	57	57	57	57	57
58	58	58	58	58	58
59	59	59	59	59	59
60	60	60	60	60	60
61	61	61	61	61	61
62	62	62	62	62	62
63	63	63	63	63	63
64	64	64	64	64	64
65	65	65	65	65	65
66	66	66	66	66	66
67	67	67	67	67	67
68	68	68	68	68	68
69	69	69	69	69	69
70	70	70	70	70	70
71	71	71	71	71	71
72	72	72	72	72	72
73	73	73	73	73	73
74	74	74	74	74	74
75	75	75	75	75	75
76	76	76	76	76	76
77	77	77	77	77	77
78	78	78	78	78	78
79	79	79	79	79	79
80	80	80	80	80	80
81	81	81	81	81	81
82	82	82	82	82	82
83	83	83	83	83	83
84	84	84	84	84	84
85	85	85	85	85	85
86	86	86	86	86	86
87	87	87	87	87	87
88	88	88	88	88	88
89	89	89	89	89	89
90	90	90	90	90	90
91	91	91	91	91	91
92	92	92	92	92	92
93	93	93	93	93	93
94	94	94	94	94	94
95	95	95	95	95	95
96	96	96	96	96	96
97	97	97	97	97	97
98	98	98	98	98	98
99	99	99	99	99	99
100	100	100	100	100	100

№	Тема	Содержание	Вопрос	Ответ	Комментарий
					Вопрос: ...
1	...	...	...	...	...
2	...	...	...	...	...
3	...	...	...	...	...
4	...	...	...	...	...
5	...	...	...	...	...
6	...	...	...	...	...
7	...	...	...	...	...
8	...	...	...	...	...
9	...	...	...	...	...
10	...	...	...	...	...

...

№	Тема	Содержание	Вопрос	Ответ	Комментарий
1	...	...	...	...	...

[illegible]

№	Наименование	Единица измерения	Формула	Условие	Комментарий
1	Время	с			Время — это продолжительность процесса.
2	Скорость	м/с	$v = \frac{s}{t}$	$v = \frac{s}{t}$	Скорость — это отношение пути к времени.
3	Ускорение	м/с²	$a = \frac{v}{t}$	$a = \frac{v}{t}$	Ускорение — это отношение скорости к времени.
4	Путь	м	$s = vt$	$s = vt$	Путь — это произведение скорости на время.
5	Время	с	$t = \frac{s}{v}$	$t = \frac{s}{v}$	Время — это отношение пути к скорости.
6	Скорость	м/с	$v = \frac{s}{t}$	$v = \frac{s}{t}$	Скорость — это отношение пути к времени.
7	Ускорение	м/с²	$a = \frac{v}{t}$	$a = \frac{v}{t}$	Ускорение — это отношение скорости к времени.
8	Путь	м	$s = vt$	$s = vt$	Путь — это произведение скорости на время.
9	Время	с	$t = \frac{s}{v}$	$t = \frac{s}{v}$	Время — это отношение пути к скорости.
10	Скорость	м/с	$v = \frac{s}{t}$	$v = \frac{s}{t}$	Скорость — это отношение пути к времени.
11	Ускорение	м/с²	$a = \frac{v}{t}$	$a = \frac{v}{t}$	Ускорение — это отношение скорости к времени.
12	Путь	м	$s = vt$	$s = vt$	Путь — это произведение скорости на время.
13	Время	с	$t = \frac{s}{v}$	$t = \frac{s}{v}$	Время — это отношение пути к скорости.
14	Скорость	м/с	$v = \frac{s}{t}$	$v = \frac{s}{t}$	Скорость — это отношение пути к времени.
15	Ускорение	м/с²	$a = \frac{v}{t}$	$a = \frac{v}{t}$	Ускорение — это отношение скорости к времени.
16	Путь	м	$s = vt$	$s = vt$	Путь — это произведение скорости на время.
17	Время	с	$t = \frac{s}{v}$	$t = \frac{s}{v}$	Время — это отношение пути к скорости.
18	Скорость	м/с	$v = \frac{s}{t}$	$v = \frac{s}{t}$	Скорость — это отношение пути к времени.
19	Ускорение	м/с²	$a = \frac{v}{t}$	$a = \frac{v}{t}$	Ускорение — это отношение скорости к времени.
20	Путь	м	$s = vt$	$s = vt$	Путь — это произведение скорости на время.

_____ , _____ . _____ .
_____ . _____ ; _____ .

2. _____ (_____) _____
_____ ; _____
_____ . _____ — _____
_____ .

3. _____ , _____
_____ . _____
_____ — _____
_____ .

4. _____ , _____ , _____ , _____
_____ (_____) , _____
_____ . _____
_____ ? _____ : _____ , _____ , _____
_____ — _____
_____ .

5. _____ . _____ " " / " _____
_____ — _____ (§ _____) . _____
_____ :
_____ . _____
_____ .

6. _____ . _____
_____ : _____ , _____
_____ — _____ , _____
_____ (_____) _____
_____ — _____ . Δ _____
_____ — _____
_____ . _____ : _____
_____ , _____
_____ .

7. _____ (/ " _____) _____
_____ , _____
_____ (§ _____) .

_____ : _____ .
_____ .

- [illegible]

_____ . _____ : _____
_____ (_____ - _____ / _____
_____ ; _____ - _____
_____).

8. _____ , _____ . _____ " _____ — _____ — _____
_____ - _____ . _____ ; _____ - _____ _____ -
_____ .

9. _____ , _____ . _____ ,
_____ , _____
_____ . _____ - _____ , _____
_____ - _____ . _____ - _____

10. _____ -
_____ (" _____ → _____ " , " _____ ") . _____
_____ ; _____
_____ .

11. _____ . _____ : _____
_____ , _____ " _____
_____ (_____ , _____
_____) _____
_____ (_____ -
_____) , _____ .

12. _____ . _____
_____ (§ _____) — _____
_____ " _____ " _____ . _____ - _____
_____ .

13. _____ . _____ , _____
_____ , _____ - _____
_____ (_____
_____) . _____
_____ .

2. Steering ● [RC BUILD]

IN PLAIN ENGLISH

The front wheels pivot on one swiveling beam, just like a toy wagon — actually, it *is* one. I salvaged the pivot assembly from a kid's wagon, and that solved the hardest mechanical problem for free. Instead of a kid pulling a handle, a strong servo motor swings the beam left and right through a "tie rod" (a threaded rod with a ball joint on each end, same idea as in your car). The Pi steers by sending the servo a timed electrical pulse; the servo moves to that angle and holds it. This part matters: the servo runs on 24 volts, and only its thin *signal* wire goes anywhere near the Pi.

Figure 2-1: The front of the car, showing the servo motor and the tie rod. The servo is connected to the Pi via a signal wire. The tie rod is a threaded rod with ball joints on each end, connecting the servo to the front wheels. The car is shown from a side-on perspective, highlighting the steering mechanism.

The servo motor is connected to the Pi via a signal wire. The tie rod is a threaded rod with ball joints on each end, connecting the servo to the front wheels. The car is shown from a side-on perspective, highlighting the steering mechanism.

1. The servo motor is connected to the Pi via a signal wire. The tie rod is a threaded rod with ball joints on each end, connecting the servo to the front wheels. The car is shown from a side-on perspective, highlighting the steering mechanism.
2. The servo motor is connected to the Pi via a signal wire. The tie rod is a threaded rod with ball joints on each end, connecting the servo to the front wheels. The car is shown from a side-on perspective, highlighting the steering mechanism.
3. The servo motor is connected to the Pi via a signal wire. The tie rod is a threaded rod with ball joints on each end, connecting the servo to the front wheels. The car is shown from a side-on perspective, highlighting the steering mechanism.
4. The servo motor is connected to the Pi via a signal wire. The tie rod is a threaded rod with ball joints on each end, connecting the servo to the front wheels. The car is shown from a side-on perspective, highlighting the steering mechanism.

The servo motor is connected to the Pi via a signal wire. The tie rod is a threaded rod with ball joints on each end, connecting the servo to the front wheels. The car is shown from a side-on perspective, highlighting the steering mechanism.

IN PLAIN ENGLISH

Each rear wheel works like a bicycle in reverse: the motor spins a small sprocket, and a bike chain carries that to a big sprocket bolted to the wheel. The big sprocket has five times the teeth of the small one, so the wheel turns five times slower than the motor — but with five times the twisting force. That's the "5:1 reduction," and it's what lets a fist-sized motor push a loaded cart through soft sand. There are two motors, one per wheel, with no axle between them — each side drives on its own. And read the orange warning below before you do anything else: the tires will just spin on their hubs unless you bond them first. This one bit me.

[illegible]

1. The first part of the document discusses the importance of maintaining accurate records of all transactions and the role of the accounting system in providing reliable financial information. It highlights the need for transparency and accountability in financial reporting.

2. The second part of the document focuses on the internal control system, which is designed to prevent and detect errors and fraud. It emphasizes the importance of segregation of duties, authorization, and documentation in ensuring the integrity of the financial data.

3. The third part of the document addresses the issue of asset protection and the role of physical and digital security measures. It discusses the importance of regular audits and the implementation of robust security protocols to safeguard the organization's assets.

4. The fourth part of the document explores the impact of technology on accounting and the role of automation in streamlining financial processes. It highlights the benefits of using cloud-based accounting software and the importance of data security in the digital age.

5. The fifth part of the document discusses the role of the accounting department in strategic decision-making and the importance of providing timely and accurate financial analysis to management. It emphasizes the need for collaboration between the accounting department and other business units.

6. The sixth part of the document addresses the issue of compliance with financial regulations and the role of the accounting department in ensuring adherence to legal requirements. It highlights the importance of staying up-to-date with regulatory changes and implementing effective compliance programs.

7. The seventh part of the document discusses the role of the accounting department in managing risk and the importance of identifying and mitigating potential financial risks. It emphasizes the need for a proactive approach to risk management and the role of the accounting department in providing risk assessment and reporting.

8. The eighth part of the document explores the role of the accounting department in corporate governance and the importance of providing accurate financial information to shareholders and other stakeholders. It highlights the need for transparency and accountability in financial reporting.

9. The ninth part of the document discusses the role of the accounting department in managing the organization's reputation and the importance of providing accurate financial information to the public. It emphasizes the need for transparency and accountability in financial reporting.

10. The tenth part of the document discusses the role of the accounting department in managing the organization's sustainability and the importance of providing accurate financial information to stakeholders. It emphasizes the need for transparency and accountability in financial reporting.

[REDACTED] [REDACTED] [REDACTED] ([REDACTED], [REDACTED]-[REDACTED]-[REDACTED]): [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED] (~\$[REDACTED] [REDACTED] ~\$[REDACTED]-[REDACTED] [REDACTED]) [REDACTED] [REDACTED] [REDACTED] — [REDACTED] [REDACTED] [REDACTED], [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]. [REDACTED]:
I [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] — [REDACTED] [REDACTED], [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] ([REDACTED] [REDACTED] [REDACTED] [REDACTED], \$[REDACTED]). [REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED].

... , \$... (... - ... , ...
...). ... —
... , ⚠ ...
... : ... /"
... # ...
... # ... (... — ...) - ... + ...
... : # ... /" (...) . -
... : ... — ... ,
... (... ° ...) . - ... : # ... -
...

★ ... — ... :
... # ... /" ...
... ; ...) .
... — # ...
... : ... → ...
→ ... — ...
... : ... / ... - ...
"..." ... —
... (...) , ...
...

4. Power system & distribution ● [RC BUILD — Buck #2 is ● autonomy-only]

IN PLAIN ENGLISH

Everything runs off one battery: a native 24-volt (25.6V nominal) 50Ah lithium iron phosphate pack. The big muscles (motors, steering servo) drink that "24V bus" straight; the delicate electronics get 5V from buck converters. Three things guard the bus: a **fuse** (melts and disconnects if a short tries to pull huge current — that's your fire protection), a **TVS diode** (a surge protector that clamps voltage spikes — I didn't know what a TVS diode was either, until a spike killed my first Raspberry Pi), and a **bulk capacitor** (a small energy sponge that smooths out jolts). And don't skip the "star ground" rule at the end: every ground wire meets at exactly one point. That keeps the motors' electrical noise out of the sensors' faint signals.

(... , ... /): ... —
... (... , ...) , ... /
... — ... (\$...) .
... — ...
... (

.../...),
... .. (...)
... ..
... ..
... ..
(...) —
... ..
... (... : ... / " ... - ... , ... / " ... - ...)
... (...) . Δ
— ... " ... §

1. +
... : ... (+) $\rightarrow$... $\rightarrow$... $\rightarrow$... $\rightarrow$
2. ... : ... (... / ... ; ... - ...
... ..
... —
...) : ... " ... ,
... " ... — ... -
— ... -
... ..
3. ... (... /) : —
... / ... $\rightarrow$ + , ... $\rightarrow$... , ... (... ; ... =
...)
... # - ... ;
... - ... , ... / . Δ ...
... :
... ..
... ..
... —
4. ... - ... ($\geq$...)
(...) .
5. ... $\rightarrow$... : ... # = ... + ... ; ... # =
... (... - ...) .
... (~ ...) .
6. ... (... - ...) : ... (... , ...)
... ..
... — ... , ... , ...
... / ...

```

// Set the motor speed to 50%
// (0 = stop, 100 = full speed)
// The motor speed is set by the PWM duty cycle.
// The duty cycle is the ratio of the pulse width to the period.
// The period is the time between the start of two consecutive pulses.
// The pulse width is the time the pulse is high.
// The duty cycle is 50% when the pulse width is half the period.
// The period is 20ms (1/50Hz).
// The pulse width is 10ms.
// The duty cycle is 50%.
// The motor speed is 50%.

```

```

// Set the motor speed to 50%
// (0 = stop, 100 = full speed)
// The motor speed is set by the PWM duty cycle.
// The duty cycle is the ratio of the pulse width to the period.
// The period is the time between the start of two consecutive pulses.
// The pulse width is the time the pulse is high.
// The duty cycle is 50% when the pulse width is half the period.
// The period is 20ms (1/50Hz).
// The pulse width is 10ms.
// The duty cycle is 50%.
// The motor speed is 50%.

```

5. Compute (Raspberry Pi 5) [AUTONOMY ONLY — the RC build has no Pi]

IN PLAIN ENGLISH

The Raspberry Pi is the cart's brain — an \$80 computer the size of a credit card, running Linux. Every sensor reading, steering decision, and safety check in this guide is a program running on it. Two things matter most right now: power it properly (clean 5V from Buck #1, never straight battery voltage) and keep it cool (a hot Pi slows itself down, and a slow brain makes a sluggish cart). The "hardware PWM" stuff below just means two of the Pi's pins can make perfectly steady timing pulses in dedicated silicon. Steering and throttle use those two pins because software-made pulses jitter — and jittery pulses made my motors twitch until I figured that out.

```

// Set the motor speed to 50%
// (0 = stop, 100 = full speed)
// The motor speed is set by the PWM duty cycle.
// The duty cycle is the ratio of the pulse width to the period.
// The period is the time between the start of two consecutive pulses.
// The pulse width is the time the pulse is high.
// The duty cycle is 50% when the pulse width is half the period.
// The period is 20ms (1/50Hz).
// The pulse width is 10ms.
// The duty cycle is 50%.
// The motor speed is 50%.

```

1. `pinMode(steeringPin, OUTPUT);` // Set the steering pin to output mode.
2. `pinMode(throttlePin, OUTPUT);` // Set the throttle pin to output mode.
3. `analogWrite(steeringPin, 50);` // Set the steering pin to 50% duty cycle.
4. `analogWrite(throttlePin, 50);` // Set the throttle pin to 50% duty cycle.

6. Motor controller (VESC Duet XS 60V dual VESC — VESC Labs)

[RC BUILD]

A brushless motor can't just be hooked to a battery — something has to fire its three wire coils in exactly the right rhythm, thousands of times a second, or it just twitches. That something is the ESC (electronic speed controller). A "VESC" is the open-source kind: same job, but you can see and tune everything from a free laptop app called **VESC Tool**. This section is almost all *settings* — teaching the controller what motor it's spinning, how fast it can go, how much current it can pull, and how to brake. I earned every number below by watching something misbehave, so copy them exactly. And watch for this: our unit is really *two* controllers in one box, and every setting has to be written to *both halves*. Forgetting the second half got me more than once.

() — ; △ —
— " —
) — :
, — .
()
) .
△ + — /
. (→ ,
→ / , → /). " , " —
.
: , + ;
, - / - ' (§)

_____ (_____): . _____
_____, _____ — _____ ' / _____.
_____ = _____ (_____ = _____; _____ — _____
_____ — _____ _____;
_____ '). . _____ (~ _____); _____
— _____ (~ _____) _____ — _____ _____
_____ . Δ _____; _____
(_____, _____). . _____: _____ (_____ —
_____ = _____ — _____), _____, _____ /
_____ (_____ — _____ ' _____). _____: _____
_____ — _____ _____ (_____
_____). _____ _____ (_____ — _____, §). . _____
_____ (_____): _____ / _____ / _____ — _____
_____. _____ _____ — _____
_____ _____ (_____ / _____, _____ —
_____): _____ / _____ — _____ / _____ (= _____
_____ — _____ _____ ' _____, _____
_____ — _____) / _____ — _____.
(_____ — _____, _____ — _____
~ _____, _____ ~ % _____; _____
_____ : _____ — _____). _____: _____
_____ _____; _____ _____ (_____ / _____)
_____ — _____ + _____ §.  _____
_____ _____ (_____ _____
_____). . _____ → _____: _____ / _____; _____
_____ (_____ — _____ — _____
_____ _____; _____
_____ . Δ _____: _____
_____ — _____
§ _____ = _____). _____,
_____ — _____ ' _____
_____, _____ ~ _____. _____
_____ (_____ — _____ / _____, ~ % _____) — _____
_____ "_____ / _____." . _____
_____.

_____ (_____): _____ ↑
_____ ; ↓ _____; ↑ _____
_____ — _____ (_____ / _____
_____ ↑ _____.)

_____ (_____): - _____
_____ (_____ -- _____) —

7. RC control [RC BUILD — see the RC-only wiring box below]

IN PLAIN ENGLISH

This is a normal hobby-store radio set: a pistol-grip transmitter in your hand and a matchbox-sized receiver on the cart. The receiver puts out three "channels" — steering wheel, trigger, and a switch — each as a timed pulse the Pi reads on its pins. The cart is self-driving by default; flip the channel-3 switch and a human has the wheel instantly. One thing to catch: the receiver speaks in 5-volt pulses, and the Pi's pins can only take 3.3. Every signal goes through a small "level shifter" board that translates. Skip it and the Pi reads garbage — or worse, gets damaged. (Building the RC-only cart with no Pi? Then there's no shifter at all — see the green box below.)

(b) (5) DPP, (b) (7)(C), (b) (7)(F); [REDACTED] + [REDACTED]
(\$[REDACTED], [REDACTED] + [REDACTED] - [REDACTED]). [REDACTED] = [REDACTED], [REDACTED] =
[REDACTED] — [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED] [REDACTED]. [REDACTED] = [REDACTED] [REDACTED] - [REDACTED] / [REDACTED] [REDACTED] ([REDACTED] [REDACTED]
[REDACTED] [REDACTED]; [REDACTED] = [REDACTED] [REDACTED]). [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]-
[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] + [REDACTED] [REDACTED] — [REDACTED] [REDACTED],
[REDACTED] [REDACTED] [REDACTED] [REDACTED].

8. Drive & steering software (drive.py / steering.py) ●

[AUTONOMY ONLY]

- $\frac{(\text{---} - \text{---})}{\text{---}}$, --- , --- = $\text{---} \pm$
 --- (---), --- .
- $\% \text{---}$ ($\text{---} - \text{---}$):
- --- / --- ($\%$)
" " —
-
: --- = --- —
.
.
- $\text{---} - \text{---}$: $\text{---} / \text{---}$ (---). $\triangle$
: --- —
.
- --- : $\text{---} \pm \text{---}$, --- , --- /
 --- .
- $\text{---} - \text{---}$ (--- , ---): --- -
(--- , ---) --- -
(--- — ---),
(---) $\rightarrow$ " --- "
- --- - --- .
" --- "
- --- .

... (... — ...)
...).

- ... (...): ... — ... , ...
...
- ... (...): ... =
... — , ... —
...
- ... (...): ... ~ ... — ... = ...
... — ... — ... — ...
(" ... ?" → ... → ...).
- ... : ... — ... & ...
(...) ... (...) ... — ...
...
- ... — ... : ... = ...
... — ... (... / ...) ...
... (= ... , ...
...) ...
... ; ...

9. Sensors & telemetry [AUTONOMY ONLY]

IN PLAIN ENGLISH

Sensors are the cart's senses. This section wires up three of them: six little laser rangefinders that watch for obstacles, a battery gauge that works like a smart fuel gauge, and a spinning laser that maps everything around the cart. Each subsection's Overview box explains its own sensor.

• ... — ...

IN PLAIN ENGLISH

... : ...
...
... — ...
...
... : ...
... "..."
... — ...
... "..."
...

- [illegible]

$$1. \text{ } \cdot \text{ } + \text{ } (\text{ } - \text{ })$$

IN PLAIN ENGLISH

The first step is to calculate the value of the expression inside the parentheses. This is done by multiplying the value of the first term by the value of the second term. The result is then added to the value of the third term. The final result is the value of the entire expression.

- The first step is to calculate the value of the expression inside the parentheses. This is done by multiplying the value of the first term by the value of the second term. The result is then added to the value of the third term. The final result is the value of the entire expression.
- The second step is to calculate the value of the expression inside the parentheses. This is done by multiplying the value of the first term by the value of the second term. The result is then added to the value of the third term. The final result is the value of the entire expression.
- The third step is to calculate the value of the expression inside the parentheses. This is done by multiplying the value of the first term by the value of the second term. The result is then added to the value of the third term. The final result is the value of the entire expression.
- The fourth step is to calculate the value of the expression inside the parentheses. This is done by multiplying the value of the first term by the value of the second term. The result is then added to the value of the third term. The final result is the value of the entire expression.

11. UWB localization — WORKING (4 anchors + tag, [4A] fix @3.3Hz) ● [AUTONOMY ONLY]

IN PLAIN ENGLISH

This is how the cart knows where you are — the heart of follow-me. UWB (ultra-wideband) is basically indoor GPS: four radio "anchors" on the cart's corners each ping a small "tag" you carry and time how long the radio wave takes to come back. Radio travels about a foot per nanosecond, so the round-trip time *is* the distance. Four distances from four known corners pin down exactly where the tag is, same as GPS satellites pin down your phone. This section has the strictest rules in the whole guide, because these little radio modules are the most fragile thing on the cart. They run on exactly 3.3 volts and die instantly at 5. They'll even kill *themselves* if powered without two tiny "decoupling" capacitors steadying their power during a transmission burst. I didn't know any of that going in — about half the modules I ever bought died teaching me. Follow the rules below and yours won't.

The following table shows the results of the regression analysis for the dependent variable Y (in thousands of dollars) against the independent variable X (in thousands of dollars). The regression equation is $\hat{Y} = a + bX$, where a is the intercept and b is the slope. The coefficient of determination R^2 is 0.85, indicating a strong positive correlation. The standard error of the estimate is 1.2. The regression line is shown in the graph below.

$$x_1, \dots, x_n \in \mathbb{R}^n \text{ (linearly independent) } \Rightarrow \det(x_1, \dots, x_n) \neq 0$$

- [illegible]

--	--	--

12. Follow-me autonomy (modules/follow_me.py) — WORKING, road-tested [AUTONOMY ONLY]

This is the program that does the actual following. Every third of a second it takes the four anchor distances, works out where you are (a bit of geometry called "trilateration" — finding the one point that fits all four measured distances), and turns that into two numbers: which way to steer (toward you) and how fast to go (based on how far away you are — it slows as it gets close and stops about 8 feet away, so it never crowds you). On top of that are the manners that make it trustworthy: it wakes up disarmed and won't move until you deliberately cycle the tag off and on, it refuses to drive at you if you're behind it, and it never, ever reverses on its own in sand — a stuck cart has to stay light enough to push out by hand.

[illegible]

... (), ... (...
& ... = ... , ...), ...
... —
... —

... (/): ...
... → ... (... - ...). ... " ... ,
... " —
...

... (... , ... - ... /): ...
... (... / , ... * ...): ...
... " ...
... " ...
... : ... , ... , ...
... " ...
... —

... (...): - ... : ... ; ...
... → ... (... → ...). ... - ... -
... (... - ...
... ' ... — ... /) - ...
... : ... / ... - ... (≤ - ...) = ... , ...
(... || ...) ...
... ; ... - ...
... ' ... - ...). - ... (\$...); ...
... -  ... (...): ... , ...
... — ... - ...
... + ... - ...
(... - ...)

... (... , ...): ... - ... - ... — ...
... - ...
... () ... ' ...

(b) (7)(C), (b) (7)(D) — [REDACTED] : [REDACTED]-[REDACTED]. [REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED] ([REDACTED]) [REDACTED] [REDACTED] [REDACTED], [REDACTED]
[REDACTED], [REDACTED] [REDACTED], [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]/[REDACTED] [REDACTED] [REDACTED]. [REDACTED] [REDACTED]
([REDACTED]/[REDACTED] — [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]). [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]: [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]. [REDACTED] [REDACTED] [REDACTED], [REDACTED] [REDACTED]

[REDACTED] [REDACTED]: [REDACTED] - [REDACTED] (M), [REDACTED] -
[REDACTED] W / M [REDACTED] (M), [REDACTED] [REDACTED] ([REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED], [REDACTED] - [REDACTED], M), [REDACTED] ⚡ [REDACTED]
[REDACTED] (M) — [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] ([REDACTED] [REDACTED]
[REDACTED] [REDACTED]). [REDACTED] [REDACTED] [REDACTED]: [REDACTED] + [REDACTED] W —
[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED].

[REDACTED] [REDACTED] ([REDACTED] / [REDACTED] [REDACTED] . [REDACTED], [REDACTED]): [REDACTED] → [REDACTED] [REDACTED] → ~/
[REDACTED] - [REDACTED] / [REDACTED] / ([REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]). [REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] - [REDACTED] . [REDACTED] — [REDACTED] - [REDACTED], [REDACTED]
[REDACTED] - [REDACTED] . [REDACTED] ([REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]).

15. Software, running & autostart ● [AUTONOMY ONLY]

IN PLAIN ENGLISH

The cart's software starts itself: flip the power on, and about a minute later everything — drive, sensors, follow-me, the radar screen, the web panel — is running with no keyboard or monitor. That's done with "services" (Linux for "always run this program, restart it if it dies, run this cleanup if it crashes"). This section lists them all, plus how to run things by hand when you're tinkering, and the automatic backups that push every change to a private online copy twice an hour.

- [illegible]

6. $\frac{1}{2} \times \frac{3}{4} = \frac{3}{8}$ — $\frac{1}{2}$, $\frac{3}{4}$, $\frac{3}{8}$; $\frac{1}{2}$, $\frac{3}{4}$, $\frac{3}{8}$ ($\frac{1}{2}$, $\frac{3}{4}$, $\frac{3}{8}$).

16. 3D-printed parts (generators in 3d_models/, PETG unless noted) ● + ● [wheel hubs + battery box = RC too]

IN PLAIN ENGLISH

Every custom bracket, spacer, and case on the cart is 3D-printed in PETG — a plastic that handles sun and moisture better than the usual PLA. The odd part: instead of drawing these parts in a CAD program, each one is generated by a small Python script. A dimension tweak is a one-line edit and a re-run. No printer? Every part here is simple enough for a local print shop or library makerspace to run from the STL files.

[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]; [REDACTED] [REDACTED]
[REDACTED] [REDACTED] ([REDACTED]/[REDACTED]) — [REDACTED], [REDACTED] [REDACTED] [REDACTED]. [REDACTED]
[REDACTED] [REDACTED] [REDACTED] [REDACTED]).

- [illegible]

- $\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.
- $\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.

$\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.

IN PLAIN ENGLISH

The voltage divider circuit is a simple circuit that takes an input voltage and divides it into two parts. The first part is the voltage across the first resistor, and the second part is the voltage across the second resistor. The voltage across the second resistor is the output voltage.

$\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.

17. Key pin & config reference (full map: PINOUT.md)

IN PLAIN ENGLISH

The one-page cheat sheet: which wire lands on which Pi pin, plus the key electrical numbers. When something's acting up or you're rebuilding, check this list first.

- $\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.
- $\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.
- $\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.
- $\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.
- $\text{V}_{\text{out}} = \left(\frac{\text{V}_{\text{in}}}{\text{R}_1 + \text{R}_2} \right) \times \text{R}_2$ — V_{out} is the voltage across R_2 , V_{in} is the voltage across $\text{R}_1 + \text{R}_2$. R_1 and R_2 are the resistances of the resistors in series.

- [illegible]

18. Open items as of 8/3

IN PLAIN ENGLISH

The honest to-do list. Every real build has one — putting it out here keeps me from pretending the cart is more finished than it is.

- [illegible]

8. $\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$ ($\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$, $\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$, $\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$); $\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$ $\rightarrow$ $\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$.
 $\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$.

9. $\frac{a^2 - b^2}{a^2 + b^2} + \frac{a^2 + b^2}{a^2 - b^2} - \frac{a^2 - b^2}{a^2 + b^2} = \frac{a^2 - b^2}{a^2 + b^2}$;
 $\frac{a^2 - b^2}{a^2 + b^2} = \frac{a^2 - b^2}{a^2 + b^2}$.

10. / : .

11. [REDACTED] (S) — [REDACTED]
[REDACTED].

[illegible]

This is the free preview — every  block is real, tested content: the exact settings, part numbers, wiring rules, and the failures that produced them. Get the Complete Build Package at kmdynamics.com