PADMA ROBOTICS PROGRAMME

A 3-Term Curriculum Framework — Grades 6–8

Awaken Innovation Lab · Padma + KanKaung

Version 2.0 · supersedes v1.0 · **Duration:** 1 academic year (3 terms × 12 weeks) · **Contact time:** ~54 hours
Platform: Padma (TurboWarp-based, Scratch 3 semantics) on Windows and Android **Hardware:** KanKaung line-tracker, ESP32

---

0. How to read this

IF YOU ARE…	READ
Awaken: sales or partnerships	§1, §3, §9
A school deciding to buy	§1.3, §9 (cost), §11 (rollout)
The teacher	§4, §5, §6, §8
Curriculum reviewer	§2, §7, and the Curriculum Map

Companion documents: the Curriculum Map (descriptor mapping), the three Term lesson packs, the Challenge Cards, and the Certificate & Transcript pack.

---

1. The product, honestly described

1.1 Padma

A visual programming editor built on **TurboWarp**, itself a Scratch 3 derivative. That lineage matters in four specific ways:

PROPERTY	CONSEQUENCE FOR TEACHING
Scratch 3 block semantics	Every Scratch resource and textbook transfers. Teachers are not learning a proprietary language.
Compiles to JavaScript	Loops run at near-native speed — a control loop can complete hundreds of cycles per second, which is exactly what a line follower needs.

PROPERTY	CONSEQUENCE FOR TEACHING
`run without screen refresh`	A routine can execute fully within one frame. This is the difference between a robot that stutters and one that glides. Taught in Term 3 Week 5.
Runs offline (Electron / Capacitor)	The offline-first promise is real, not a workaround.

> Padma is not Scratch. It programs like Scratch because it shares the lineage, but the compiler, hardware bridge and packaging are Awaken's. Teach it as Awaken's tool; use Scratch vocabulary because that is the language students meet everywhere else.

1.2 KanKaung

COMPONENT	SPEC	CURRICULUM ROLE
Controller	ESP32	The "process" in input → process → output
Drive	2 × continuous-rotation servo	Differential steering. Speed by pulse width, not duty cycle.
Line sensing	2 × IR reflectance	The core control input. Two sensors is the minimum that supports a **decision** rather than a switch.
Proximity	1 × HC-SR04 ultrasonic	Distance, threshold, hysteresis, obstacle response
Expansion	**8 free GPIO**	**Club only.** Outside the assessed curriculum.
Safety	**Firmware watchdog**	Motors stop if commands cease. Confirmed present; it is load-bearing for Term 1.

Why two IR sensors is the right number. A one-sensor follower can only say "on the line" or "not on the line." A two-sensor follower can say **which side** — left, right, centred, or lost. That one extra bit turns a switch into a controller, and it gives students a four-row truth table that becomes the spine of the whole year.

1.3 What's in the box, and what the school must add

IN THE KIT	SCHOOL MUST SUPPLY
KanKaung chassis, 2 servos, 2 IR, 1 ultrasonic, ESP32	Tablet or Windows laptop per 3–4 students
Wheels, battery holder, firmware with watchdog	Batteries (consumable, replaced every lesson in Term 3)

IN THE KIT	SCHOOL MUST SUPPLY
Padma licence	**Line mats** — white board + black electrical tape, made by the teacher
Pairing instructions	Spare servos (they strip gears)
—	Storage boxes, one per kit

The line mat is not a product. 60 × 90 cm of white foam board with 19 mm black tape. Six mats in an hour for the price of a roll of tape. Say this loudly in every sales conversation — a curriculum that depends on a proprietary mat dies when the mat is lost.

1.4 Purchase quantity

Four kits per class of 32, not eight. The Term 1 and Term 2 robot sessions rotate half the tables at a time, which halves the hardware line on the quote without halving the experience. Every student drives KanKaung four times across the year.

That is a sales point, not a compromise: a school that could not justify eight robots can justify four.

1.5 Six facts to confirm before classroom print

Written from the product brief and standard ESP32 practice. Each is a one-line change if different; none changes the structure.

1. **Exact block names** in Padma's hardware extension (motors, IR, ultrasonic).
2. **Connection method** — Bluetooth serial on Android, USB serial on Windows, or both.
3. **Battery chemistry and voltage** — determines the safety rules in §10.3.
4. **Servo neutral point and direction convention.**
5. **IR logic** — does black read `1` or `0` on shipping units? This flips every conditional in Term 3.
6. **Which 8 GPIO are free**, and which are input-only.

---

2. Design principles

1. **Remote → Rules → Autonomy.** Term 1 the student decides. Term 2 the student writes the rules. Term 3 the machine decides. Every lesson fits this arc.
2. **The robot is present from day one, programmable from Term 3.** It sits on the teacher's desk in Term 1, drives under student control in Weeks 4 and 6, and is programmed in Term 3. Presence builds anticipation; delay builds competence.
3. **Concept before artifact.** Students write the four-state table in Term 2, on paper, for a machine they have not met. Term 3 Week 4 runs it. Two months of anticipation, one lesson of payoff.

4. **Two sensors, four states, one real controller.** No simulated sensors. IR is cheap, fast and honest — it shows students exactly what they got wrong.
5. **Speed is a trade-off, not a number.** A fast robot overshoots; a slow one is stable. That trade-off is the lesson, not an obstacle to it.
6. **The kit outlives the course.** Eight free pins mean the same robot is a platform for the club, next year, and the science fair.
7. **Everything works offline.** No accounts, no cloud, no connectivity assumptions anywhere.

2.1 The rotation model

Four kits, eight tables, half the class drives at a time.

TERM	ROBOT WEEKS	TABLES 1–4	TABLES 5–8
1	4, 6	Week 4	Week 6
2	5, 7	Week 5	Week 7
3	every week	all	all

Non-driving tables are never spectators. They get a specific observation job — "watch the left wheel, tell me when it stops" — recorded on a slip. Watching without a task is corrosive by the fourth week.

---

3. Program outcomes

By the end of Term 3, every student can:

- **PO1** Build a Padma project using events, loops, selection, variables and custom blocks.
- **PO2** Drive KanKaung from a student-written remote-control interface.
- **PO3** Explain the four-state logic of a two-sensor line follower, on paper, without the robot.
- **PO4** Calibrate two drive servos and explain why the values drift over time.
- **PO5** Implement a control loop and tune it for stability at a chosen speed.
- **PO6** Diagnose a robot fault by classifying it as code, wiring, power or surface.
- **PO7** Document a build so another team can reproduce it.

---

4. TERM 1 — SIGNALS

"You drive it."

Narrative. The robot is in the room from day one, switched off. Everything learned this term is a tool needed to wake it up.

WK	FOCUS	NEW CONCEPT	ROBOT
1	Meet Padma, meet KanKaung	Program, hat blocks, events	on the desk
2	Precision drawing	Sequence, parameters, angles	—
3	The stage as a map	Coordinates, heading	—
4	**Mission control**	Events, broadcast	**tables 1–4 drive**
5	Start, stop, state	State machine	—
6	**Loops**	`repeat N`, `repeat until`	**tables 5–8 drive**
7	Nested loops	Pattern, the lotus tile	—
8	Composition	Three constructs at once	—
9	Debugging studio	Six-step protocol	—
10	Build sprint	Decomposition	—

Week 4 is the hook. Four sprite buttons drive a real machine. The test: hand the device to another team's Navigator and say nothing — if they can drive it without asking, the interface works.

Week 6 is the loop made physical. `repeat until <not pressed?>` — the first time a loop in this course is responsible for something moving.

Week 8 has no hardware. The robot stays on the desk. The lesson is *"that machine has to do exactly this, four hundred times a second, with nobody watching."*

Term 1 project — Mission Console. A Padma app with four event-driven controls, a status display driven by a variable, a nested-loop pattern, and a written operator card.

---

5. TERM 2 — RULES

"You teach it to decide."

Narrative. Everything built this term is the robot's brain, written before it has a body.

WK	FOCUS	NEW CONCEPT	ROBOT
1	The decision problem	Conditions, measurable facts, **AND / OR / NOT gates**	—
2	If / else	Branches, comparisons	—
3	**Compound conditions**	`and` / `or` / `not` + **the four-state table**	—
4	Variables	Set, change, clamp	—
5	**Variables that matter**	Trim, calibration	**tables 1–4 drive**
6	Modes	State machine	—
7	**Functions**	Custom blocks, parameters	**tables 5–8 drive**
8	Parameters	Generalising routines	—
9	Integration	Contracts, message bus	—
10	Testing	Boundary values, hostile input	—

5.1 Week 3 is the hinge of the year

The Week 1 foundation. Week 1 introduces the three gates explicitly — AND, OR and NOT — with a four-row truth table for each. Week 3 then reads `and`, `or` and `not` as those gates, so the four-state table arrives as a truth table the class has already met, one input wider.

Students fill in the four-state table by hand — for a machine they have not met.

LEFT IR	RIGHT IR	MEANING	ACTION
on	on	centred	both forward
off	on	drifted right	slow left
on	off	drifted left	slow right
off	off	lost	stop or search

Do not let them code it. Paper only. The temptation to "just try it" spoils the Term 3 reveal and teaches nothing extra.

Collect the tables and file them. They are the Term 3 Week 4 lesson. Losing them costs a week.

5.2 The two robot sessions

Week 5 — the trim bench. Two servos never match; one is about 4% faster. Over a metre that is four centimetres of curve. Students tune `trimL` / `trimR` until drift is under 3 cm, changing one value per run.

> *"Your robot does not drive straight. That is not a fault. That is manufacturing tolerance, and every real robot has it."*

Week 7 — the countdown. `announce(n)` on the robot. One routine, three starting values. The first parameterised block with a physical consequence.

5.3 Term 2 project

A decision-making console with shared resources, modes, reusable routines and a written test suite — plus the four-state table, hand-written, which is the artifact the year hangs on.

---

6. TERM 3 — AUTONOMY

"It drives itself."

Narrative. Two terms of rules on a screen. Now the rules run on four wheels and a floor that does not care how clever the code is.

> **This term assumes Term 2.** Conditionals, variables, modes and functions are not re-taught. If a class cannot fill in the four-state table from memory in Week 1, that is a Term 2 problem, not a Term 3 one.

WK	FOCUS	BUILDS ON	ROBOT
1	Meet KanKaung	T1 W4	motor, trim checked
2	**Trim, properly**	T2 W5	drift explained
3	What the sensors see	T2 W1–2	truth table verified
4	**Four states**	**T2 W3**	**their own table, running**
5	Variables and loop rate	T2 W4–6	smooth follower
6	Tuning	T2 W10	fast, reliable
7	The second sense	T2 W2	obstacle stop, hysteresis
8	Priority and fail-safe	T2 W6	guardian behaviour
9	Debugging a physical system	T1 W9	fault log
10	Build sprint	everything	**course completed 8/10**

6.1 Week 2 is not "what is trim"

They learned that in Term 2. This is *"why is the number you measured in October wrong now?"* — battery sag, heat, wear.

> *"Trim is a snapshot of one moment's hardware, not a property of the robot."*

6.2 Week 4 is the payoff

> *"Nine weeks ago you filled this in for a machine you had never seen. Today that machine is on your table and this is the program. Read your own handwriting. Is it right?"*

6.3 The course

60 × 90 cm mat, black tape: 40 cm straight · 90° curve · 30 cm straight · **8 cm gap** · stop box.

The gap is where the robot goes blind for half a second — the first time it needs a **memory of duration**, not just a sensor reading.

6.4 What is not in this term

The 8 free GPIO. They belong to the club (§12.2). Keeping Term 3 tight is what makes the course completable in ten weeks.

---

7. Assessment

Four pillars, four levels, used weekly.

PILLAR	WHAT IT MEASURES
Logic & Problem Solving	Decomposition, logic gates, correct control structures
Code Efficiency & Cleanliness	Loops, functions, naming, readability
Debugging Perseverance	Method, one-change-at-a-time, recovery
Creative Collaboration	Roles, explanation, building on ideas

Term grades: 50% culminating project · 25% engineering notebook · 15% checkpoint practicals · 10% showcase defence.

Never average the four pillars. A student at ③ Logic and ① Debugging needs a different intervention from the reverse.

Term 1 now carries hardware evidence. Week 4's remote-control interface is assessed live, not by screenshot. The Debugging pillar gains a physical descriptor: "*distinguishes a code fault from a loose lead.*" No fifth pillar — a fifth column on a table a teacher is holding while standing up is a fifth column nobody reads.

Record keeping. The Curriculum Map defines 27 descriptors and which week secures each. The Certificate & Transcript pack turns those into a defensible end-of-year record.

---

8. Teacher enablement

8.1 The realistic teacher

Most will not have written code before. Design accordingly.

- **Three-day bootcamp** before the year: day 1 Padma from zero, day 2 Padma plus KanKaung, day 3 both together with a fault-injection lab.
- **Every teacher must drive KanKaung in month one.** If Term 3 is where they first connect hardware, the Term 1 robot weeks will already have exposed the gap.
- **Every lesson has the same six-part shape** — mission log, unplugged concept, build-along, studio, debug minute, exit ticket.
- **A laminated debug card per table** covering the twelve faults that consume most lost time.

8.2 The fault matrix

Every one of these looks like a code bug. Almost none of them is.

SYMPTOM	LIKELY CAUSE	CHECK
Spins in place	Left/right motor commands swapped	Swap the two pins
Curves constantly	Servo mismatch	Trim calibration
Follows 2 s then stops	Battery sag	Fresh cells
Ignores the line entirely	IR logic inverted	Read both sensors
Loses it on curves	Loop too slow	`run without screen refresh`
Wobbles violently	Speed too high for loop rate	Reduce speed, re-tune
Stops at the gap	Expected	Coast logic, Week 10
One wheel dead	Servo lead or stripped gear	Swap from spares
Works on the desk, not the mat	Surface reflectivity	Matte tape, no gloss

8.3 Pacing guardrails

IF YOU ARE…	TERM 1	TERM 2	TERM 3
1 week behind	Merge W6+W7	Shorten W6	Shorten W6
2 weeks behind	Cut W7, keep W9	Cut W8 → fold parameters into W7	Merge W2+W3
3 weeks behind	Convert W11 into the only build sprint	Cut to 2 subsystems	**Do not cut.** Run a shorter course.

Never cut: the Term 2 Week 3 table, the Term 3 Week 4 lesson, the debugging weeks, the showcase.

---

9. Cost model (indicative — confirm with Awaken)

LINE	PER CLASS OF 32
4 robot kits (rotating model)	Awaken pricing
8 tablets or Windows laptops *(if not owned)*	$800–1,600
Batteries (consumable)	$60/year
Line mats — 6 boards + tape	$40
Spare servos (4)	$12
Printed cards, lamination	$80
Marginal cost, year 2+	**~$200**

Four kits, not eight. A single class set serves every grade 6–8 section in sequence and does not expire at the end of the course.

---

10. Operations

10.1 Room setup

- One table per team, 4 students, robot in a numbered box

- One charging station, labelled
- A **driving area** taped on the floor for Term 1 and 2 sessions — 1.5 × 1 m, bounded
- Mats stored flat, not rolled

10.2 Device management

- Padma installed from a local APK/installer; no accounts, no store
- One device per team, numbered to match the robot
- **Pair before the lesson, every time.** A pre-paired robot is 30 seconds of setup; pairing in front of 32 children is fifteen minutes of your lesson gone.

10.3 Safety

1. **Robot on the mat or in the taped area, never on a desk edge.**
2. **Power off before touching wheels or wires.**
3. **Batteries to Awaken's specification.** If lithium, charge attended and never overnight; if AA, replace rather than recharge.
4. **Servos get hot and strip gears.** Do not hold a wheel while it drives.
5. **Cap speed at 40 for Term 1 remote control.** Full speed ends in stripped gears and a frightened class.

10.4 The three failures that will happen

- **Pairing fails** → spare robot, log it, move on. Never debug pairing for more than 2 minutes.
- **A servo dies** → swap from spares. They are consumables, not durable goods.
- **The mat is damaged** → tape is cheap. Keep a roll in the room.

---

11. Rollout plan

WHEN	ACTION	OWNER
−12 wk	Confirm the six facts in §1.5 against shipping product	Awaken
−10 wk	Order **4 kits** per class + spares	School
−8 wk	Select two teachers; book bootcamp	School
−6 wk	Build and test one complete kit end-to-end	Awaken + lead teacher
−4 wk	Three-day bootcamp; every teacher drives the robot	Awaken trainer
−3 wk	Dry run with 6 volunteer students	Lead teacher
−2 wk	Revise; print cards; build mats	Lead teacher

WHEN	ACTION	OWNER
0	Term 1 begins	—

The dry run matters most. One afternoon with six students surfaces more than three days of planning — particularly around pairing, trim, and how long mat-building actually takes.

---

12. Appendices

12.1 Vocabulary (wall poster)

Program · instruction · sequence · event · trigger · loop · nested loop · variable · conditional · state · function · parameter · sensor · actuator · threshold · hysteresis · trim · calibration · deadband · closed loop · GPIO.

12.2 The club

The 8 free GPIO live here, outside the assessed curriculum. Six sessions, one project each:

1. Status beacon — LED and buzzer
2. Obstacle rover — second ultrasonic, no line
3. Sumo — two robots, one ring
4. Line-maze solver — three-way junctions, memory of turns
5. Sensor logger — temperature and humidity, no motion
6. Gripper arm — mechanical part required, advanced only

The club is the renewal path. Club students become next year's Term 3 teaching assistants, which is how the programme scales without hiring.

12.3 What this framework deliberately does not do

- It does not simulate the robot. Simulation teaches the block palette; only the robot teaches control.
- It does not teach the GPIO expansion in curriculum time. That is the club's job.
- It does not teach text-based code. That is the Bridge to Text, a separate product.
- It does not assume internet, cloud accounts or app stores — anywhere.

12.4 Companion documents

DOCUMENT	PURPOSE
Curriculum Map	27 descriptors, mapped lesson by lesson
Term 1–3 lesson packs	30 lessons, full scripts

DOCUMENT	PURPOSE
Challenge Cards	12 studio mini-projects
Certificate & Transcript	Issuance pack and wording rules

Awaken Innovation Lab · Padma Robotics Programme framework v2.0 · Six product facts in §1.5 must be confirmed before classroom print.