Robotics Engineering Track Syllabus

Robotics Projects Track is an integral part of the Storming Robots learning ecosystem, supported by:

- Structured level progression with rigor (shown in this syllabus)
- Clear assessment methodology to verify mastery.
- Proven pathway and long-term development model
- Built-in competition readiness benchmarks
- Instructor enablement: training + guardrails for consistent delivery

1)CONTENTS

About this Track

All of Storming Robots programs are grounded in true engineering principles, guiding students through a structured and immersive engineering process—distinct from many general "robotics" programs that are more play-oriented.

Storming Robots' Robotics Projects Track is a long-term engineering pathway, not a one-time camp or a single-season robotics class. Students' progress through a structured level system with clear standards, consistent rigor, and assessment-based placement. As they advance, they build real engineering habits: systematic design, testing, troubleshooting, and algorithmic problem-solving—supported by text-based programming and computational thinking.

Unlike many programs that focus on short-term builds or fixed "cookie-cutter" competition setups, our track is designed for measurable growth and true competition readiness. Each level is supported by proven learning benchmarks, documented progress, and instructor enablement (training + guardrails) to ensure consistent quality and outcomes. The result is a reliable pathway from beginner to advanced robotics—building the foundation students need to excel in STEM and beyond.

Characteristics

1. **Clear, level-based engineering pathway (not standalone classes)**
    - Students' progress at their own pace and are not confined to a single term.
    - Advancement occurs only after demonstrated mastery, ensuring rigor and continuity across terms.
2. **Problem-solving driven by computational thinking and text-based programming.**
    - Focused on real automation and engineering logic, not drag-and-drop graphical robotics.
3. **Strong analytical skill development**
    - Learning is built on reasoning, method, and systematic problem-solving—not rote memorization.
4. **Mathematics used as a thinking tool as complexity increases.**
    - Students apply logic, modeling, measurement, and optimization as projects mature.
5. **Higher-order thinking embedded in daily work.**
    - Minimal spoon-feeding: students learn to design, infer, test, and troubleshoot with discipline.
6. **Consistent assessment of predictable outcomes**
    - Quality control that supports reliable advancement, especially at higher levels and for competition readiness.

7. **Instructor guardrails that ensure consistency**
    - Standards, level-gating, and assessments prevent learning quality from depending on a single instructor's style.
8. **Ecosystem advantage: Robotics + Computer Science under one roof**
    - A unified pathway that supports deeper automation, stronger foundations, and long-term growth.
9. **Open-source, highly expandable platforms**
    - Engineering freedom with significantly lower cost than many proprietary ecosystems.

.

Pedagogical Focus

1) **Engineering cycle:** Design → Prototype → Testing → Iteration
2) **Core skill priorities:** computational thinking, algorithmic reasoning, and precision debugging.
3) **Professional practice:** engineering journals, design presentations, and technical documentation
4) **Progression model:** structured labs early on, then a deliberate shift to open-ended projects by Level IV+

How to do Well in this Track?

Best Practices:

1) Design with Flowchart.
2) Follow the Foundations of Engineering Process along with Computational Thinking
    - Consistently developing the habits that strong engineers rely on.
    - Breaking large problems into manageable steps.
    - Designing solutions using flowcharts.
    - Divide and Conquer approach.
3) Log your engineering journal.
4) Complete Exercises Thoughtfully:

— Stay inquisitive and self-disciplined — not just know how to implement the solutions but acquire the analysis process to arrive at the solutions. Do not allow yourself to be satisfied just because a program set seems to be complete, but only after you feel you do understand.

— Thoughtful Tests and Trouble-shooting systematically (not just blindly trials and errors)

5) Indispensable process of analysis:

— The cornerstone of true learning is to go through the process of analysis independently. It's important to engage deeply with a problem by breaking it down, experimenting, and learning from mistakes; NOT merely implementing a given solution.

— Limit acquiring external assistance without putting effort into trying to solve problems. Receiving substantial external assistance at home to complete assigned work often will quickly cause you to miss out on the critical process of analysis. This indispensable learning element strengthens your ability to apply in various scenarios.

— Stay perseverant in taking on challenging problems. Ensure that you first try to work through a problem on your own, even if it means struggling a bit, before seeking outside help.

6) Avoid Rote Memorization:

— Simply memorizing a technique without understanding the underlying logic will leave you unprepared when the problem is presented in a new or slightly tweaked form. Strive to understand the 'why' and 'how' behind each method.

7) Practice, Reflect, and Iterate:

— This is a journey—complete with mistakes and adjustments. It is essential to develop strong analytical skills that you can apply in any future more advanced activities such as exams, competitions, and projects.

8) Systematic trouble-shooting - trial and errors, not just wild guess.

Danger of Receiving too much External Assistance

Relying on external help before fully engaging in your own analytical process can hinder your development in critical ways:

1) Lack of Self-Reliance: Early dependence prevents you from cultivating the confidence to solve problems on your own.

2) Shallow Engagement: Without digging deep into the problem, you miss out on understanding the underlying principles.

3) Lack of Critical Thinking: Simply implementing given techniques, without analysis, means you won't develop the ability to think critically on your own.

4) Limited Creativity but just being passive: Over-reliance makes you a mere do-er instead of someone who can innovate and adapt when problems change.

Bottom line: The **indispensable process of analysis is the cornerstone of true learning**. True mastery comes from working through problems independently making mistakes, learning from them, and developing your own problem-solving strategies. While help from sources like Stack Overflow or family members can be beneficial, it should only complement your journey, not replace it.

Learning Resources—

https://learn.stormingrobots.com

www.tinkercad.com

Software required

- Visual Studio Code
- Platformio

Honor Code

Programming assignments are pledged work and are bound by the honor code. To simply put, the violation may be in any of the following forms:

- Claim someone else's work as their own.
- Edit someone else's work by simply changing variables or style, etc., and claim to be the result of your own work.
- Complete assigned work with substantial help from others to the extent that you cannot even explain your own work.

Violation will disqualify you from participating in any competitions with Storming Robots.

Hardware for Level B to IV:

SR's **custom platform** (< $200 - market-dependent**)** is a one-time purchase that supports at-home practice and **Levels B–IV**, offering far more versatility than $400+ LEGO. Storming Robots source and verify retail components from multiple suppliers to save families time, shipping, and ensure accuracy for students.

Despite the fact we use it for mainly Levels B to IV, platform will continue to be usable at college level.

When to purchase and take home: At the beginning of Level B: Robot kit is kept at the center initially; no charge until students are ready to take it home, then we'll share the bll of materials for parents' reference.

For Level V-VI: There will be *incremental* upgrades of about $100 (market-dependent). This approach lets the robot expand and grow rather than be replaced.

~~~~~~ LEVEL B ~~~~~~

Covered Concepts:

- Learn to utilize Tinkercad as a start (with no need for physical hardware).
- Basic Usage Guide for Inhouse Repository System with Arduino Microcontroller.
- Students begin by developing the habits that strong engineers rely on
- Construct base robot kit.
- Explore rudimentary programming logic and motions.

To conclude Level B:

At the end of Level B, students should be able to:
- Program robots to do basic motions.
- Simple display with OLED.
- Simple object detection with ultrasonic sensor.
- Simple B/W recognition with light sensor.

- Simple primitive variables.

~~~~~~ LEVEL I ~~~~~~

Covered Concepts:

- Understanding basic conditional programming logic - single level of logical structure
- More into display with OLED
- Design with Flowchart (a single level of conditional.
- Learning more about how to utilize sensors to make intelligent decisions:
    - Utilize a single-color sensor for line tracking.
    - Utilize a single ultrasonic sensor as proximity sensor but using a time of flight as a more accurate object detection – gain the idea how they are different.
    - Utilize pixy or buzzer for signaling purpose.
    - Button for simple control purpose.
    - Simple usage of serial monitor.

To conclude Level I:

- Be able to design with simple flowchart to clearly show the control structure to help programming control with 2 sensors at the same time.
- Understand pros and cons using ultrasonic sensor vs time of flight.

~~~~~~ LEVEL II ~~~~~~

Covered Concepts:

Learn to manipulate more complex logic with multiple sensors.

"Nested control structure" – that's where most students find the most challenging.

More complex display with OLED.

More complex design with Flowchart.

May explore simple modularization with simple void function.

Rudimentary usage of Color sensors to recognize colors (R | G | B).

Explore using Servo motor.

To conclude Level II:

At the end of Level II, students should be able to:

- design flowchart – from high level to lower level (modules) on their own with some supervision.
- Utilize a servo motor to control for actuator like a flap, a single arm for a single-axis motion.
- Handle projects requiring knowledge of "nested control conditions."

~~~~~~LEVEL III ~~~~~~

Covered Concepts:

- More in-depth Modularization handling – functions with parameters and returning data.
- Move beyond single-sensor logic to coordinate multi-sensor decision-making, enabling their robots to handle higher-level, competition-style challenges with greater reliability.
- More advanced OLED Display
- Rudimentary level in using motor encoder.
- Understanding basic gear math – translate distance traveled to encoder and vice versa.

To conclude Level III:

At the end of Level III, students should be:

- Design flowchart from high level to broken down lower level with only some assistance.
- Conduct projects more complex than level II including using gear math for more accurate distance traveled.
- Improving upon handle more complex nested control structure.

~~~~~~ LEVEL IV ~~~~~~

Covered Concepts:

1) Introduction to Arrays (Foundational Data Structures)

- Not just essential, but pivotal concept.
- A challenging milestone for younger students, arrays teach how to store larger sets of data in an organized way and retrieve information efficiently.
- An essential step toward algorithmic thinking and future AI-oriented work.

2) Using Multiple Color Sensors for Complex Tasks

3) Understanding the Binary World

4) Reinforcing Reusable Functions & System Design

Knowing how to use data array is not only fundamental, but pivotal before starting to handle proper decision making for high school level robotics projects.

To conclude Level IV:

At the end of Level IV, students should be able to:

- handle complex projects requiring data array to store data; and how to use the data element for robot decision making and navigation.
- Determining what data, a function should receive and what it should return.
- Using flowcharts to design multi-module programs with clearer input/output relationships between components

~~~~~~ LEVEL V ~~~~~~

Covered Concepts:

- Learning to use more native Arduino libraries stop using more high-level abstracted functions calls provided for Level B to IV.
- Learning to use Inertia Measurement Unit
- More in-depth understanding using Serial Monitor
- Learning to use multiplexer

Special Note to Grade 9 or someone who has already good amount of knowledge using Arduino?

For students who join SR at 8th or 9th grade who come in with prior robotics and programming background, we offer a fast-track approach so they can move through the Level B-IV stages efficiently and transition into full Level V work as soon as they are ready. This is because they will need to fill any remaining foundational gaps—for example: constructing the robot platform, understanding the underlying mechanical math, following a proper engineering process (including flowcharts before coding), setting up a consistent repository/workflow between home and the center, and reinforcing core programming skills as needed.

To conclude Level V:

By the end of Level V, students should have improved structured thinking which inherently helps them to create ***scalable, maintainable*** programs rather than one-time solutions.

This group will be prepared to tackle the challenging RoboCup pre-college Division Robotics Entry -level Competition.

Reference:

- RoboCup – AI-oriented robotics competition: pre-college – RoboCupJunior

~~~~~~LEVEL VI ~~~~~~

Prerequisite:

Must have completed Algorithms in C/C++ - Level I or equivalent – see the CS syllabus.

Proficiency in Algebra I and Geometry.

Covered Concepts:

PICO Breadboard Design with fundamental understanding in electronics. Such as:

- Analog vs Digital pins. How analog pins act like digital pins.
- Pull-up vs Pull down resister.
- Potentiometer
- Voltage Divider
- How to utilize photoresistor and a few other forms of resistors
- Deeper knowledge of I2C, wiring, memory addressing, the distinction between data and clock lines.

Advanced navigation, state-machine design, and feedback control; exposure to algorithmic decision-making and early path-planning concepts.

Introduce computer vision for color and object detection.

To conclude Level VI:

This group will be prepared to tackle the challenging RoboCup pre-college Division Robotics higher tiered Competition.

COPYRIGHT