
# Task Sheet

## 28th February - 1st March 2026

ICSS
Imperial College Space Society

x

LODESTAR

# Topic

The rapid growth of space debris in Low Earth Orbit poses a significant threat to operational satellites and future missions. Whilst LEOHack 2025 explored the engineering challenges of actively removing debris from orbit, **with SatHack, we aim to introduce you to the equally urgent problem of satellite collision avoidance.**

# Tasks

Over the course of this weekend, **you will be challenged with 2 Tasks**

1. **An Embedded Systems Physical Task**
2. **A Conceptual Mission Design Task**

Every team competing should attempt **both** tasks. They will be scored separately, and then scores will be combined to determine the winner of the hackathon. Both tasks are equally weighted, in that they both have a maximum score of 250 points.

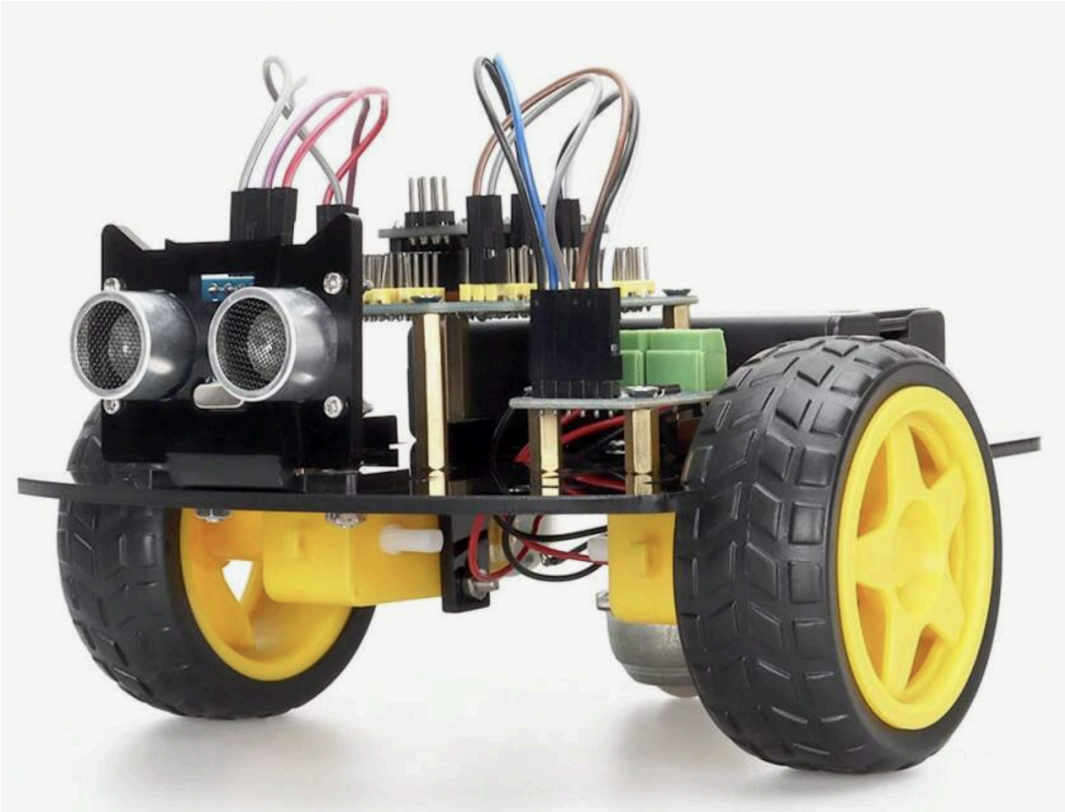
More details on both tasks and their respective scoring systems are given in the rest of the document.

# Table of Contents

# Embedded Systems Task

**The aim of this task is to program and assemble an autonomous sensing and navigation system to successfully move a robotic ‘satellite vehicle’ through a variety of  ‘debris fields’ without any collisions.**

The satellite will be represented by an **RC car vehicle** equipped with an **ultrasonic sensor** for sensing of the environment and an **Arduino Nano** on which all sensing, decision-making, and control must be performed. The **simulated debris field** will consist of cardboard boxes, foam cylinders, balls, and transparent perspex sheets that the robot will have to expertly navigate around. **More details on both of these are given below.**

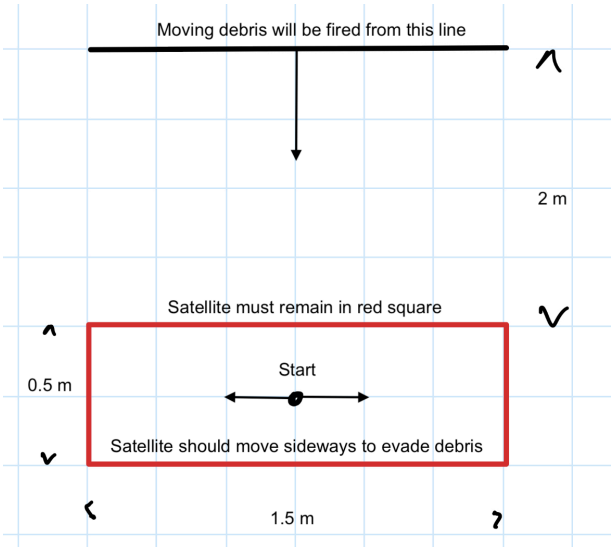


*The provided RC car vehicle that serves as your ‘Satellite’*

## Task Specifics

As described above, over this weekend you will be tasked with programming a provided rc car robot to navigate a variety of debris fields. More specifically, your satellite vehicle will be challenged to autonomously navigate **3 obstacle course stages simulating progressively complex debris scenarios**.

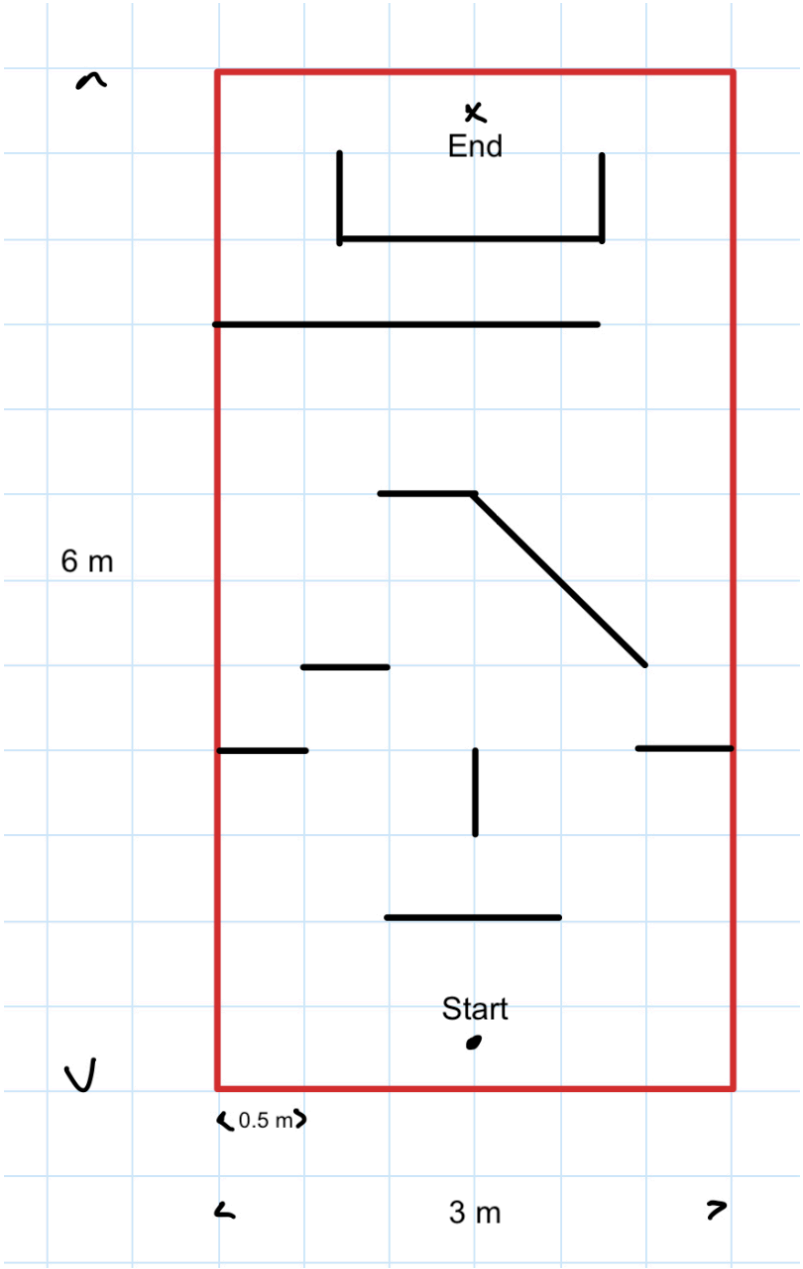
- **Stage 1 "Space Invader-Like" Evasion**: The robotic "satellite" moves laterally* to dodge incoming debris in real time.

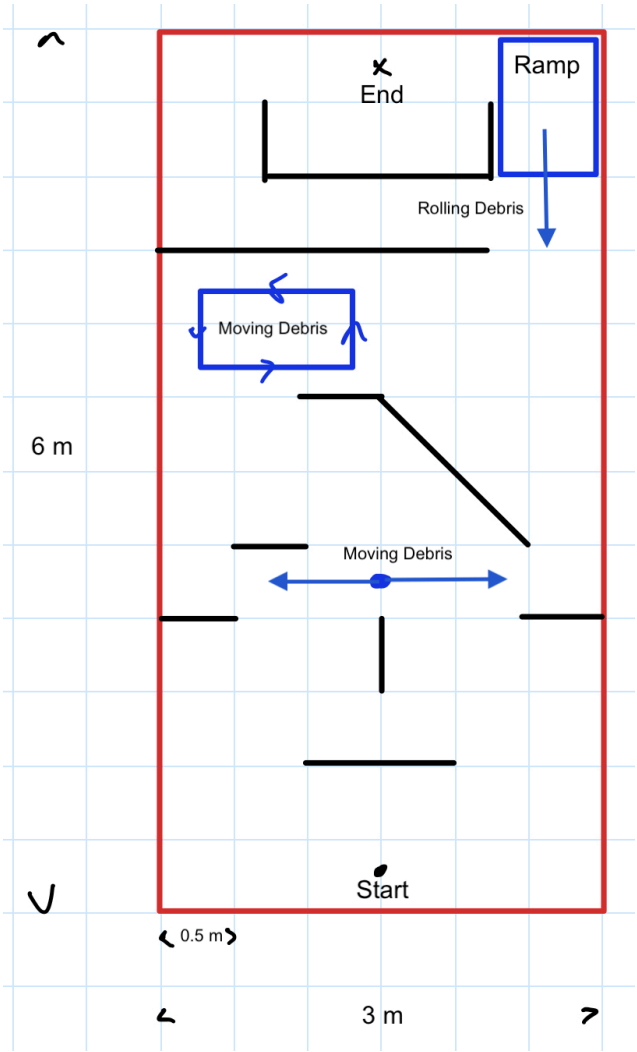

Debris will be launched approximately every **5** seconds, roughly in the direction of the satellite's current position. The satellite should aim to evade the incoming debris for **30** seconds for a completely successful test (see the scoring criteria for more details).

*Note: the provided rc cars cannot move 'laterally' in the sense of left-right motion while the sensor points at the oncoming debris in the standard setup. **We recommend either: placing the sensor on the side of the rc vehicle for this task**, or, **triggering a 90 deg rotation before moving 'laterally' upon detecting oncoming debris, then turning 90 deg back the other way to face the debris field again.**

- **Stage 2 Static Debris Field**: The robot must navigate a static debris field, starting in the specified start position and finishing in the end position. The red rectangle marks out the testing area and the black lines represent ‘space debris’ (largely these will be cardboard walls). Scoring will be based solely on reaching the end **without any collisions with the debris or edge of the testing area**. A run will be ended if the satellite contacts any debris and points will be gathered for the distance reached. Another run can then be attempted within the allotted testing time. (**Full scoring details are given in the submission and scoring details)**



- **Stage 3 Moving Debris Field**: The robot completes a similar course, but with moving obstacles which will test the control system's robustness in a dynamic environment. Stage 3 builds on the static debris field of stage 2, but with notable additional dynamic obstacles. An RC car representing space debris will move linearly covering the lower gap below. A second RC car will trace a loop further up the course as shown. Finally, a ramp will be built in the upper right corner from which rolling debris will be launched at a rate of once every 10s.




**Note: all of these setups will only first be set up on the day of the hackathon itself, so they may be subject to change should any of the setup elements be practically infeasible! If this is the case, we will notify all of you! (This is particularly the case for Stage 3!)**

Details on when you can use the debris fields for testing etc. are given in the Submission, Testing and Scoring details section.

## What You're Given

**Physical Apparatus**

- 1 RC Car Kit including,
    - 1 HC-SR04 Ultrasonic Sensor
    - 1 Arduino Nano
    - Motor Drivers
    - Motors
    - Battery Case
    - Jumper Wires
    - 2 Tyred Wheels and 1 Universal Wheel
    - Body + Attachments
    - Hardware (nuts bolts etc.)
    - Screwdrivers
    - USB A to C Cable for programming of Arduino from a personal laptop
    - Instruction Leaflet

- 2 Batteries
- Shared hacking consumables
    - Further Jumper Wires
    - Electrical Tape
    - Zip Ties
    - Glue
- **Potentially** spare HC-SR04 Ultrasonic Sensors if given out sensors are found to be faulty, but this **cannot be guaranteed.**

**Demonstration Software**

Some of the ICSS team have taken the time to play around with the robots and write a demo script to work off of, hopefully making the initial setup of the robots easier. **Note the robot set also comes with helpful documentation**

**https://github.com/sonic597/sat-hack2026/new/main**

Participants will also be provided access to plenty of professional expertise from the Lodestar Space employees attending.

## Submission, Testing and Scoring Details

**At 3 PM on Sunday, you will be asked to stop work on developing your control systems (we will ask for your Arduino's to be submitted at 3 to ensure so).**

Following this, testing will be carried out on each team over the 3 sequential stages of simulated debris fields.

For each stage, multiple details of a team's performance will be noted down to determine a score:

- **For Stage 1 testing, the following scoring criteria will be used**

| Criteria and Penalties | Allocation |
|---|---|
| Time until first collision | Points = time lasted in seconds |
| Completion of full 30 second test with no collisions | 20 (Additional*) Points |
| Penalty for leaving the testing area | -15 Points |
| Any external interaction with the test from the testing team | Run is disqualified |
| Any external interaction from others (other than those releasing debris) | Restart run with time regained |

*So if a run were to survive the full 30 seconds without any collisions, it will be scored 50 points (the maximum for this test)

- **For Stage 2 and 3 tests, the following scoring criteria will be used**

| Criteria and Penalties | Allocation |
|---|---|
| Distance reached before first Collision | Points = Number of Checkpoints Reached * 10 |
| Completion of the course with no collisions | 50 (Additional) Points |
| Any external interaction with the test from the testing team | Run is disqualified |
| Any external interaction from others (other than those releasing debris) | Restart run with time regained |

Hence the **Maximum Score Available is 250 Points**.

**For the final testing runs, 2.5 minutes will be allowed for attempting Stage 1, 5 minutes will then be allowed for attempting Stage 2 and 5 further minutes will then be allowed for attempting Stage 3**

## Additional Testing

**Note that testing can also be carried out at earlier points during the hackathon,** to bank some performance points earlier if desired.

However, due to the nature of the task, and the possibility of multiple people testing out the courses at the same time, it is clarified below as to what classifies as 'Additional Testing' and what time restrictions have been placed on this.

- **If a team is attempting any of the Stages and would like their run to be timed, scored and officially adjudicated, they must have an appropriate member of the ICSS committee or Lodestar present. -** For this type of testing, **each team will be allocated a maximum of 2 15-minute sessions each day.**
    - **However**, on the second day of the hackathon, a waiting list will quickly form for these testing opportunities, so be sure to book in for testing as soon as you think you might be ready for the next available slot.
    - **Similarly,** as the hackathon only runs for 6 hours on the second day, **we cannot guarantee both sessions will be available for all teams on Sunday.**
        - **Furthermore, if a team who has not tested at all yet on Sunday requests testing, they will be given precedence over waiting teams that have already tested once that day.**

- If a team just wishes to try out a chosen part of any of the stages, or attempt an unscored run, they are free to do so as much as they would like, within reason/with consideration for the following rules.
    - **A team should not hog a single element of the course for an extended period of time** (it is up to the discretion of the ICSS committee/Lodestar team to decide on this). Teams can continue to test on this element of the course, but must take turns/allow for other teams to test here as well.
        - This particularly relevant to the start of the course
    - **No elements of the testing setup/debris course can be removed from the testing area and used for testing in team spaces** (this is because we may run out of certain testing elements/struggle to track them down and retrieve them when needed, and so benefitting some teams over others)
    - **Any team using the course in this casual sense, should be expected to leave the course if another team elects to use the course for a dedicated 15-minute testing session** (described above).

# Mission Design Task

## Focus

The physical robot challenges are intentionally local in scope:
*sense → avoid → move.*

This mission design task zooms out to the systems level, forcing teams to think like space autonomy engineers. It emphasizes:

- Decision-making under uncertainty
- Safety and fault handling
- Limited communication with ground
- Resource constraints
- Verification and trust in autonomous behavior

The goal is not to design hardware in detail, but to understand how autonomy enables complex space missions where human intervention is limited or impossible.

---

## Mission Brief (Reference Mission)

You are designing an autonomous spacecraft mission to remove orbital debris from low Earth orbit (LEO).

The spacecraft must operate with limited ground supervision and make safety-critical decisions onboard.

**Scenario: Deorbit a space object (satellites, upper rocket stages etc) that is slowly tumbling in LEO SSO.**

The target is non-cooperative and was not designed to be serviced.

---

## Mission Success Criteria

A successful mission must demonstrate the ability to:

- Identify and track the target object
- Safely rendezvous and match relative motion
- Perform a controlled capture or attachment
- Execute a deorbit or disposal maneuver
- Avoid creating additional debris
- Maintain spacecraft safety under communication delays or outages

---

## Constraints

Your design must explicitly consider the following constraints:

- Limited ground contact windows.
- Communication latency, ranging from seconds to minutes depending on orbit and geometry
- Finite propellant / $\Delta v$ budget
- Sensor uncertainty, including lighting conditions, glare, occlusion, and noise
- Hard safety constraints, such as:
    - No collisions with the target or other objects
    - Restricted thruster usage during proximity operations

The spacecraft must be capable of operating safely even when ground intervention is unavailable.

---

## Your Task

**Design a concept-level autonomous mission capable of meeting the objectives above. Assume you are in charge of the spacecraft only and will be dropped off into SSO orbit by a launch vehicle (Spacex Transporter Missions).**

**You are not expected to provide detailed engineering designs or code. Focus instead on capabilities, structure, and reasoning.**

---

## Part 1 – Mission Definition

Describe:

- The overall mission goal
- The major mission phases from deployment to disposal
- What constitutes mission success vs mission failure
- The primary risks and constraints that shape mission design

---

## Part 2 – Hardware Capability Requirements

**Design a vehicle capable of carrying out the mission described.**

Begin by identifying and classifying the **on-board systems** required for successful execution of the mission.

Outline the **hardware capabilities** needed to enable this mission, including the sensing and actuation required to support autonomous operation.

For each system category:

- Describe the capability it provides to the spacecraft
- Explain why this capability must exist onboard to enable vehicle operation
- Identify the major component options within that category and briefly discuss the tradeoffs involved in selecting between them

For example:

- The spacecraft requires a **propulsion system** to perform orbital maneuvers and disposal actions
- Consider what types of propulsion are realistic for spacecraft of this class
- Discuss the tradeoffs between available propulsion options and how they impact mission capability

---

## Part 3 – Software & Autonomy Architecture

**Outline how the spacecraft would operate autonomously by describing:**

- How it perceives its environment
- How decisions are made onboard
- How unexpected situations or failures are handled

Break this down into logical software modules or layers.
Diagrams (block diagrams, flowcharts, state machines) are encouraged.

---

## Part 4 – Autonomy & Safety Considerations

**Discuss:**

- Inputs into the autonomy module
- Explain how your design is safety oriented
- How the system behaves when sensor data is uncertain or conflicting
- What actions the spacecraft takes when confidence is low
- How autonomy is tested and validated before launch

---

## Part 5 – Δv and Resource Awareness

**Space missions are fundamentally constrained by available Δv. Your design should demonstrate awareness that Δv is a finite, mission-limiting resource.**

In your slides:

- Identify the mission phases that consume Δv
- Provide Δv estimates for those phases (clearly state assumptions)
- Explain how remaining Δv influences:
    - Mission sequencing and timing
    - Autonomous decision-making
    - Abort, retreat, or continue logic under uncertainty

This is the area to express any unique features or operations that come into play in this mission.

---

### Deliverable

**A short (Maximum length of 8 minutes) video presentation of a slideshow covering:**

- Mission overview
- Hardware capability breakdown
- Software and autonomy structure
- Safety and fault - handling philosophy
- Δv awareness and resource-driven decision - making

---

## What We're Looking For

- Systems-level thinking
- Clear trade-offs and assumptions
- Realistic treatment of autonomy and safety
- Awareness of uncertainty, irreversible decisions and costly consequences

**There is no single correct solution. We are ultimately looking for teams to show us their unique solutions and get a sense for what mission design is in the real world. The scenario is left vague on purpose as to not box into a specific design - please make your best educated assumptions.**

---

## Marking Criteria

| Section | Completion of the Task (/20 Points) | Clarity of Presentation (/10 Points) | Consideration of Scenario (/10 Points) | Originality (/10 Points) |
|---|---|---|---|---|
| Part 1 | | | | |
| Part 2 | | | | |
| Part 3 | | | | |
| Part 4 | | | | |
| Part 5 | | | | |

Hence the maximum mark for this task is also **250 Points.**

# Timetable

Same as in email, but included in case useful

| Time | Event |
|---|---|
| Day 1: 0930 | Opening and Registration |
| 1000 | Talk from Lodestar Space |
| 1030 | Introduction + Outline + Q&A |
| 1100 | **Beginning of Hackathon** |
| 1300 | Lunch |
| 2000 | **End of Access to the Venue for Day 1** |
| Day 2: 0900 | **Beginning of Venue Access for Day 2** |
| 1300 | Lunch |
| 1500 | **Submission + Live Testing** |
| 17:30 | Closing Ceremony |
| 17:45-18:00 | Dismantle Components and End of Event |

## And Thank you to our Sponsors!

LODESTAR

MAGDRIVE

onshape®

SKYRORA

SoftInWay

ACCU

Ansys