2 Project Plan

2.1 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project. At minimum, this section should have a task dependence graph, description of each task, and a justification of your tasks with respect to your requirements. You may optionally also include sub-tasks.

- Identify which type of RC car we would like to modify for this project.
- Determine which sensors need to be added/changed for project criteria.
- Research how a computer could communicate with the RC car using various signals.
- Create a bridge connecting the RC car to the computer and/or host.
- Ensure a secure method of communication between the car and the computer is made.
- Build software to handle information being sent from the RC car to the PC.
- Create autonomous features based on the software previously built.
- Thoroughly test both the RC car and software prior to the final race.

2.2 Project Management/Tracking Procedures

Which of agile, waterfall or waterfall+agile project management style are you adopting. Justify it with respect to the project goals.

Will use a waterfall+agile approach to manage our project. This project management style works well with the system we'll use to track our progress over the year. This management style will also make tracking tasks through the project lifecycle easier and more efficient for the team.

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

We will track our goals using Gitlab. This will make creating, editing, and viewing tasks and a ToDo list much easier and straightforward.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

What are some key milestones in your proposed project? It may be helpful to develop these milestones for each task and subtask from 2.1. How do you measure progress on a given task? These metrics, preferably quantifiable, should be developed for each task. The milestones should be stated in terms of these metrics: Machine learning algorithm XYZ will classify with 80% accuracy; the pattern recognition logic on FPGA will recognize a pattern every 1 ms (at 1K patterns/sec throughput). ML accuracy target might go up to 90% from 80%.

In an agile development process, these milestones can be refined with successive iterations/sprints (perhaps a subset of your requirements applicable to those sprint).

- Identify which type of RC car we would like to modify for this project.
    - Select either a standard RC car, DuckyTown car, or advanced Caterpillar car.
- Determine which sensors need to be added/changed for project criteria.
    - Acquire cameras, visual/distance sensors, light sensors, frequency sensors, etc.
- Create a bridge connecting the RC car to the computer and/or host.
    - Know which radio frequency to transmit and which computer program to use for connectivity.
- Ensure a secure method of communication between the car and the computer is made.
    - Create cyber security measures to prevent signal hacking.
    - Make a secure method for reconnection in case signal is lost at any time.
- Build software to handle information being sent from the RC car to the PC.
    - Obtain values from sensors and store them for later processing.
    - Create functions to send information for turning, acceleration, etc.
- Create autonomous features based on the software previously built.
    - Use information stored from the car to make decisions and call functions for action.
    - Ensure old test values are not stored which could mess with autonomous functionality.
- Thoroughly test both the RC car and software prior to the final race.
    - Improve the efficiency of autonomous functions to a 90% success rate.
    - Ensure any user input can quickly be inputted into the car.

2.4 Project Timeline/Schedule

• A realistic, well-planned schedule is an essential component of every well-planned project

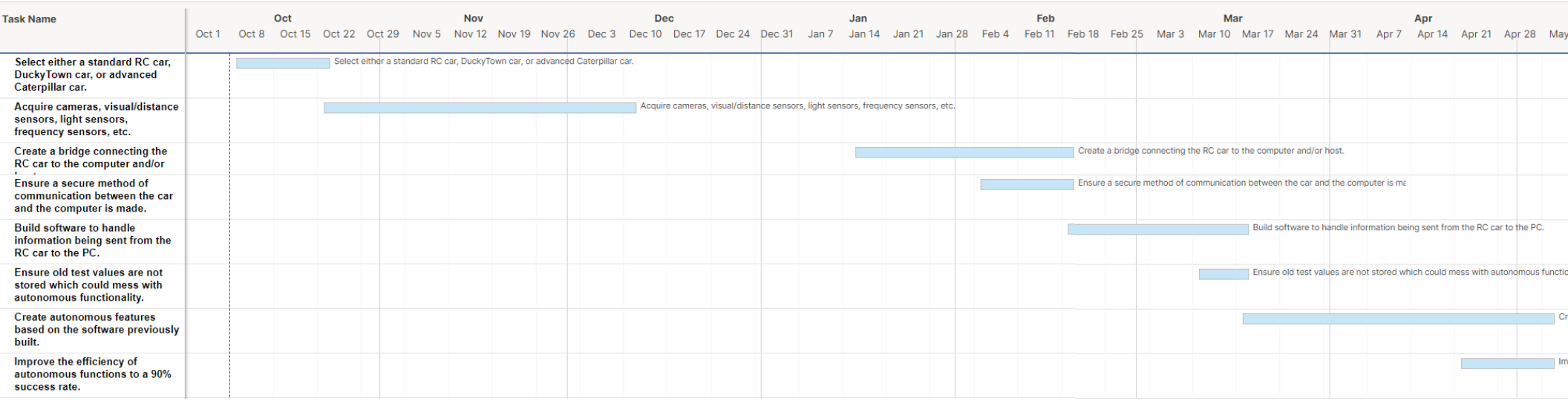
• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Project schedule/Gantt chart can be adapted to Agile or Waterfall development model. For agile, a sprint schedule with specific technical milestones/requirements/targets will work.



Tasks in order:

- **Select either a standard RC car, DuckyTown car, or advanced Caterpillar car.**
- **Acquire cameras, visual/distance sensors, light sensors, frequency sensors, etc.**
- **Create a bridge connecting the RC car to the computer and/or host.**
- **Ensure a secure method of communication between the car and the computer is made.**
- **Build software to handle information being sent from the RC car to the PC.**
- **Ensure old test values are not stored which could mess with autonomous functionality.**
- **Create autonomous features based on the software previously built.**
- **Improve the efficiency of autonomous functions to a 90% success rate.**

2.5 Risks And Risk Management/Mitigation

Consider for each task what risks exist (certain performance target may not be met; certain tool may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

Agile project can associate risks and risk mitigation with each sprint.

Now that we have gotten a general idea of the requirements and constraints for our project, the main risk we face is to build a functional RC car by the end of the year. This includes designing and implementing some changes to a preexisting RC car that the whole team can agree on. The current risks we can identify are taken from section 2.3 above, and the possible solutions are subject to change if we can come up with a better idea at the time we come across them.

Risks:

- Disagreement between which RC car we should use as a base (0.5)
    - This can be solved by using communication between the team, and if there is a split, we can use a voting process as a last resort
- Which parts should we purchase to be implemented, and how many parts to buy in case some break or malfunction (0.75)
    - Can be solved by conducting research on the parts we need and only purchasing the minimum amount we need to test and implement them. If some parts take longer to get, we can purchase them earlier so we are never stagnant in our development.
    - This will also need good budget management for our expenses so we don't go over the limit set.
- Making sure our implementations are up to the teams standards (0.6)
    - This can be done through testing what the RC car can handle, physically and electronically, while also identifying possible risks we may encounter during the final race.
    - Ideas and communication between the team play an important role in this risk, as all team members must be honest on how they feel about the current state of the project, and willing to bring forward their ideas on how to make the project better.

As of now those are the main risks we will most likely encounter while completing our project, but there is no doubt other issues may occur. When that happens, we will work as a team to generate ideas and solutions to those problems to ensure that all members of the group are satisfied with the final build of the project.

2.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.

Task (taken from section 2.1):	Expected person-hours to complete:
1. Decide on RC	2
2. Determine Sensors	4
3. Research computer interface	4
4. Create bridge connection	12
5. Ensure secure communication	16
6. Build information handling software	28
7. Create autonomous features	40
8. Thoroughly test the product	40+

1. Determining the RC car used in our project will be a process completed in conjunction with the other race teams at out weekly meetings. A decision should be reached within the course of a single meeting.
2. Determining the sensors we will use in our project will be a relatively simple discussion once again taking place with the other groups determining our joint parameters for the race to come.
3. The computer interface research can take place in conjunction with the decision on sensors, once we know the vehicle base we will be using for our project. This is solely an information gathering phase, and as such it will be briefer than the following development.
4. Establishing the bridge connection will involve programming from the appropriate wireless libraries, then a short period of debugging to ensure the connection functions properly.
5. Ensuring secure connection will take longer than establishing the connection. It will involve rigorous testing of edge cases to ensure that a potential attacker cannot gain a foothold in our communications by taking input outside of the expected use cases.
6. Developing the information handling software will require all previous tasks in the project to be completed. It will likely involve every member of the group working on individual components of the whole before combining these parts into a single piece of software.
7. Creating the autonomous features for our vehicle will be one of the two largest tasks of our project. This will involve a combination of development by our computer/software/cybersecurity engineers, and development of a physical component for our prototype.
8. The final task of our project will be stress testing, trial runs, and bug fixing. This task will occur in conjucnction with the development of our autonomous features, to ensure the project ends in a successful prototype, and it will continue for as long as our group has

time- thus, the allotted hours are subject to more variance than the other, easier to predict, tasks.

2.7 Other Resource Requirements

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

This project will require:

- An RC Car body
- A remote control
- 3D printing access for custom parts development
- A stable IDE compatible with the code used in our RC vehicle
- Mechanical components for any vehicle enhancements-
    - screws
    - axels
    - wheels
    - bearings
    - and other such parts
- Sensors for any vehicle enhancements-
    - Infrared Sensors
    - Radar Sensors
    - Sonar Sensors
    - Rotational Sensors