EC03 - Smart Tripod
Task 1

Electronics Club

June 10, 2022

Contents

Congratulations again on becoming a part of Electronics Club 22-23! Always remember that the club is not just about Electronics, it is a community that is bound together by people sharing a common love for technology! It is a place where you can grow, experiment, learn, and not fear failure.

This is one of the quotes that has stuck with me ever since I encountered it!

Sometimes it is the people no one imagines anything of who do the things that no one can imagine
— Alan Turing

1 General Instructions

Every time we provide a task, we will add in as many references as possible to make it easier for you, but remember that most of the things are just a Google search away.

Try out as many alternatives as possible before you contact any of the cores for doubts, this will make you more empowered and independent in the process.

Every time you proceed to take up a task, we suggest you ponder on the need for it (i.e., **"Why?"**) and understand the process while you follow any course/blog/tutorial online (i.e., **"How?"**). You have best understood a subject or task when you can explain it to others in the most simple words ever.

This task is divided into **4 Mini-Tasks**. We did these tasks in our PM tenure too and this gave us a huge head start for all the later tasks and projects!

Deadlines:

Mini Tasks: 19th June
Project Task: 22nd June

ALL THE BEST!

Subscriptions

Here are few technical websites which provide some food for thought on the current technical trend ongoing

- How-To-Geek
- Newsletter Science X
- Weekly Robotics
- How-To-GeekHackaday Editors

2 Mini Task 1

2.1 Learning Outcomes

- Learning Git and GitHub
- Using Command Line interface for Git
- Markdown syntax

2.2 Tasks

1. Create a GitHub account using your smail
2. Learn Git and Git bash by following any Youtube tutorial
3. Clone your project repository and create a branch with your name.
4. Create a Repository named "Task-1" using Command Line **(CL)** in your branch. Create a folder with Mini-Task 1. Add this PDF into the folder using the Command Line.
5. Add all the upcoming tasks in the same repository in a different folder.
6. Fork this repository.
   Refer Clone vs Fork to learn more

3 Mini Task 2

3.1 Learning Outcomes

- **Habit of Documentation:** Documentation is not only to retain your tasks for the future, but it will also help you track your progress and quantify the work you do and of course it's a way of organizing your workspace.
- **Explore different domains in and around Electronics:** Electronics is a huge field and envelops a myriad of domains. Try to get a taste of everything so that later you can choose from what's in the store, for you!
- **Learn different approaches to problem-solving** There are usually plenty of ways to tackle an issue. You start off with the dumbest way of course and through the process you'll learn to optimise your approach towards problem-solving.
- Get to understand different electronic components, compare them and understand how they go into different applications.

3.2 Task

Select **7 distinct projects** based on the below mentioned criteria and understand them thoroughly (If asked anything about it, you must be able to explain it clearly).
Some helpful websites: Instructables, Hackster.io, Hackaday, randomnerdtutorials, etc.

- They should strictly be from different domains
- The project should be of medium-hard difficulty to replicate.
- At Least 4 projects out of 7 should not include Arduino (Electronics is more than just building things around Arduino).

Now, document all the 7 projects on GitHub. Create a folder named "Mini-Task 2". Add a brief description of all the 7 Projects and your personal notes, comments or any improvements you can think of. Note:

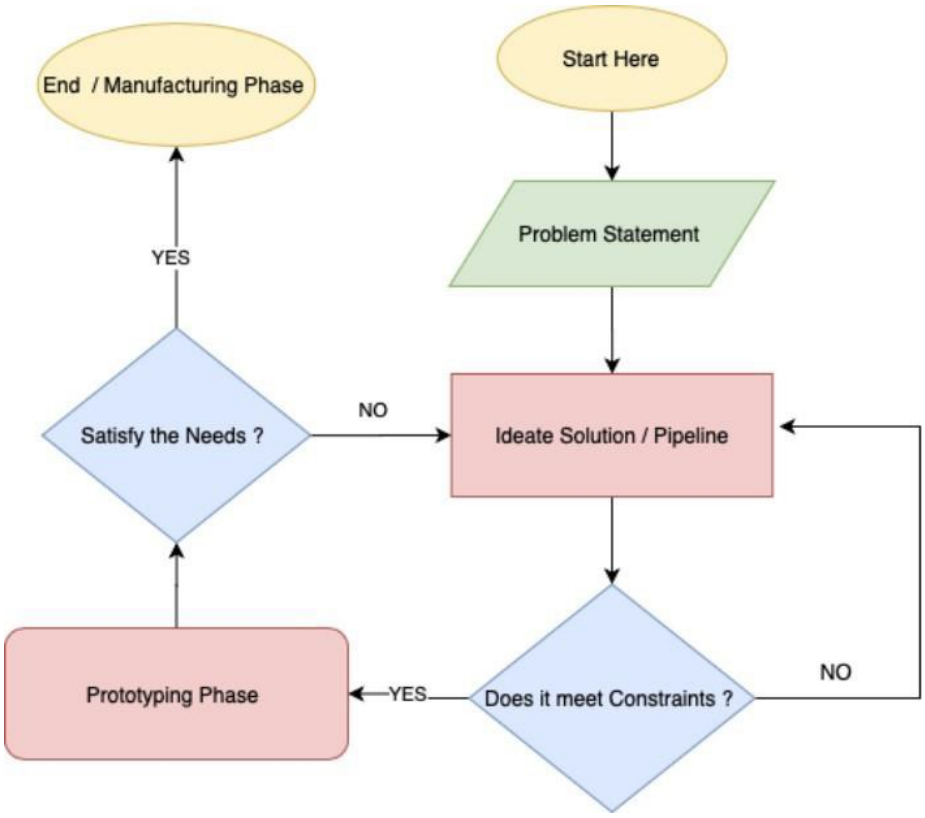
- For At Least 3 Projects, Add Images along with the description.
- For At Least 2 Projects, Add GIFs along with the description.

Example Projects

1. Password Based Door Lock System using 8051 Microcontroller
2. Gesture Controlled Wheelchair
3. Raspberry Pi Security Camera

4 Mini Task 3

4.1 Flow of a Project



This is how a project progresses over time:

- **Identify Goals - The Problem Statement**
  Defining the problem statement, goals and objectives clearly and the practicality of the project (feasibility checks) are the first things to start with. Later in the course, you may add more features, but without a well-defined problem statement, you cannot start on your project journey!

  If you define the problem correctly, you almost have the solution

  — Steve Jobs

- **Ideation and Planning**
  Pareto Principle, the **Golden Rule** states that 80% of outcomes result from 20% of all causes for any given event. Therefore, I cannot stress enough the importance of prioritizing the 20% of the factors that will

either salvage or drown the project at hand. This is where you will spend a lot of time identifying and ideating solutions to overcome the real-life hurdles such as size, budget, resources, time constraints, etc.

- **Prototyping Phase**
  This is the phase we all would be waiting for! When you finally get your components and connect all the wires and code the microcontrollers and get it to work. Lots of time (80%) you wouldn't get it working in the first attempt, then you should follow a feedback mechanism and revisit the defined constraints or jump to debugging accordingly. (P.S: I put 80% to sound optimistic, but honestly, it's a 100%)

- **End / Manufacturing phase**
  Either the project ends here if it was meant to be a learning module, or it moves into the manufacturing phase, where more ideation and more constraints are laid out and the same problem is solved and mass-produced.

4.2 Example Project

An example for this I picked from here would be:

Problem Statement: Designing a Voice controlled Bot that is responsive to following commands - 'RIGHT', 'LEFT', 'FORWARD', 'BACKWARD', 'STOP'

Ideation and Planning: Understanding the Car's Mechanism:

- Smartphone App -¿ Bluetooth Transceiver -¿ Microcontroller -¿ Motor Driver Motors
- Part of the pipeline to break:

Part of the Pipeline	**Feasibility**	**Advantages**	**Disadvantages**
Smartphone App	Hard to code an app.	Easily available for many PANs	Security issues; can be hacked easily
Bluetooth Transceiver	Easy to use SPP; widely used also	Low power consumption. Better range than IR. Cheap in terms of cost.	Low bandwidth compared to WIFI. Short range communication.
Arduino/ Microcontroller	Usually hard, unless very skilled and experienced	Easily programmable, as it has many libraries to work with.	Limited executions. Hard for beginners.
Motor Driver and Motors	Easy to control. They can be easily customized!	Works on smaller scales of Voltage and current. Variety of options available.	H-bridge ICs come in different datasheets, making it a little difficult task. Cost is fine.

Choosing a Pipeline:
From the Above, we can choose Bluetooth Transceiver or Motor driver Motors for improvements.

Bluetooth Transceiver - Here, they have chosen the HC05/HC06 Bluetooth transceiver. The difference is just master-slave operations, with up to 10m sensitivity circle.

Motor drivers - These listen to low voltage powered Arduino and control an actual motor which needs high input voltage. L293D/ L298N can be the options. Cannot use a 3.3V powered Microcontroller when the Motor Driver uses 5V, and if chosen, then need to take care of some specific things → *More Ideation on approach?*

Next, we need to draft a list of components required for the project, get them and start working.

Prototyping Phase: This is the phase to build the Bot, Test and Debug it. Some earlier background assumptions we have might fall apart here. This can also be fixed by going back to the ideation phase and working it out in a recursive fashion.

Each and every Project can be broken down like this and made better. These are also intuitively highlighted in the tutorials you follow for the first task.

4.3 The Task

Choose at least 2 projects from the previous 7 project list and suggest better approaches to solve the same problem and argue why your idea is more feasible/easy to implement /cost-effective /less power consuming, etc.

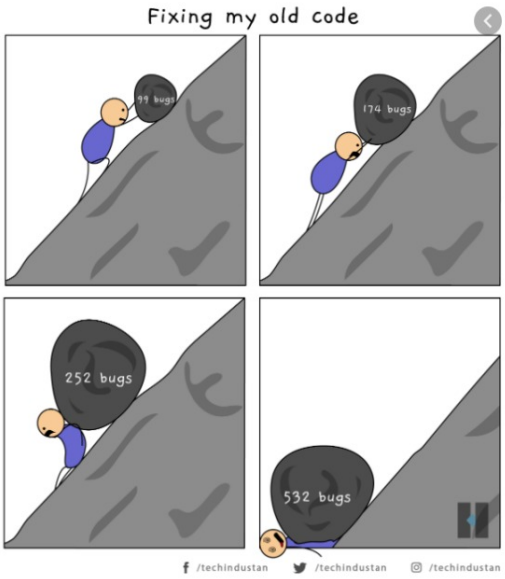
Submission format: Draft it the same way or in a better way than what has been done above. Create a folder again for this task.

5 Mini Task 4

Since nothing (majorly) works absolutely correctly in the first go, debugging is an essential part of any project. We try to rectify the errors and remove the bugs in the system and the process of doing so is called debugging.
Interesting read: Origin of a Bug

Debugging is unique for every project. Breaking a big project into chunks and verifying each and every chunk helps, similar to adding print statements after every milestone in some code - **Divide and Conquer.**

One of the famous bug stories: Windows Calculator Bug!. Here are some memes which we found funny:



5.1 The Debugging Process

This is a general process to follow:

- Identifying the error – It saves time and avoids the errors in the prototype. Doing this after every stage helps minimize errors later and also saves time.

- Identifying the error location – From the pipeline in Mini-Task 2, it becomes easier to identify the location and cause of the error.
- Analyzing the error – To understand the type of bug or error and reduce the number of errors we need to analyze the error. Solving one bug may lead to another bug that stops the application process.
- Prove the analysis – Once the error has been analyzed, we need to prove the analysis. Writing or using test cases through the test framework greatly helps and makes it full proof.
- Cover the lateral damage – The bugs can be resolved by making the appropriate changes and move onto the next stages of the project to fix the other new errors.
- Fix and Validate – This is the final stage to check all the new errors, changes and finally deploy.

Here is a debugging tutorial that our Ex-Head wrote for one of the sessions he had conducted: Link

5.2 The task

Choose any one of the same 2 projects that you chose in Mini-Task 2 and write a Debugging tutorial for it. Explain it in detail and in a recursive fashion.

Submission format: Document the explanation in a new file under the same repository.

6 Project Task

Hope you all have enjoyed the process till now! Now let's implement whatever you have learnt on the main project itself.

Visit the repository of your project and open your branch (Additional Task: Learn about branching in GitHub). Append a file into your branch and start documenting on your project. Follow the steps from Mini-Task 2 and 3. Use your Project Member app (pipeline question) for your reference.