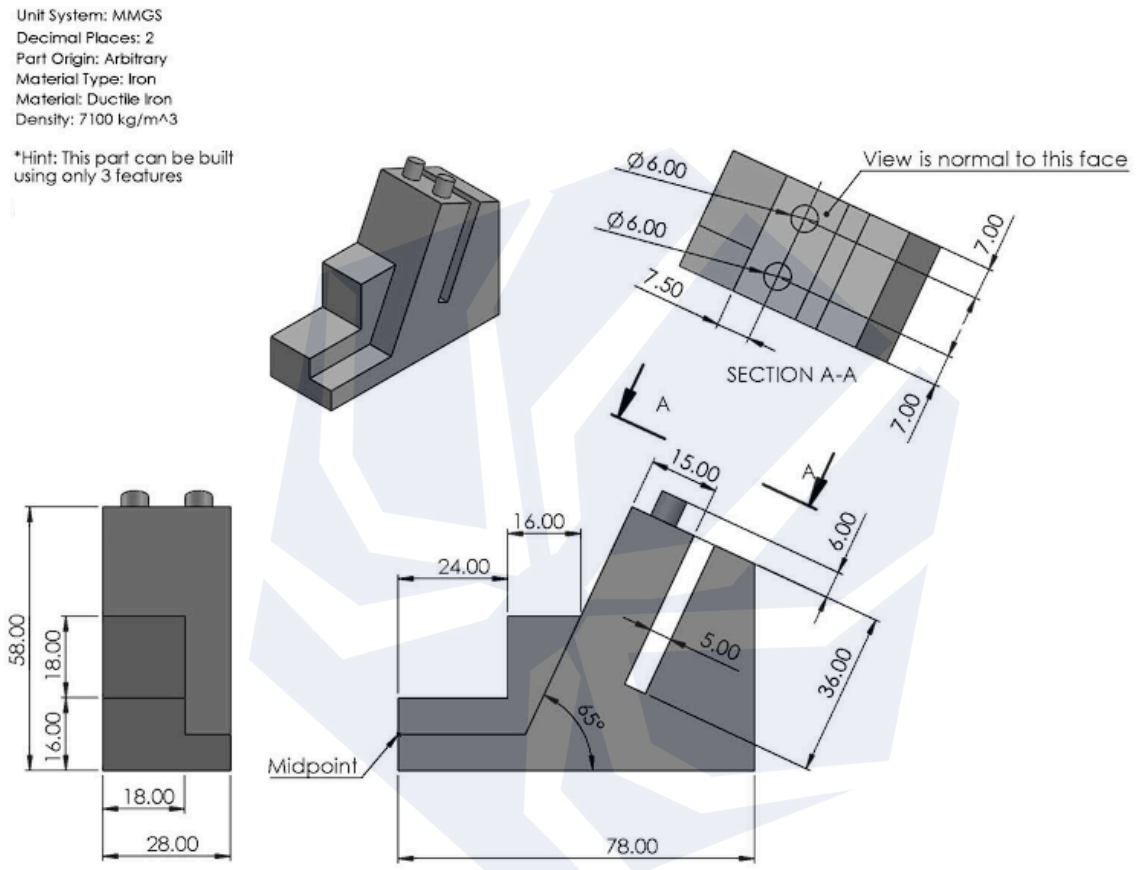
Robotics Track

Spider
R&D Club

Solid Modelling - Task 1

Task

Build the following part using the appropriate dimensions



Note:

- ➢ The units, material and density of the part is given in the top left corner

Judging Criteria

- Accuracy of mass
- Accuracy of dimensions
- Features, methodology used

Resources

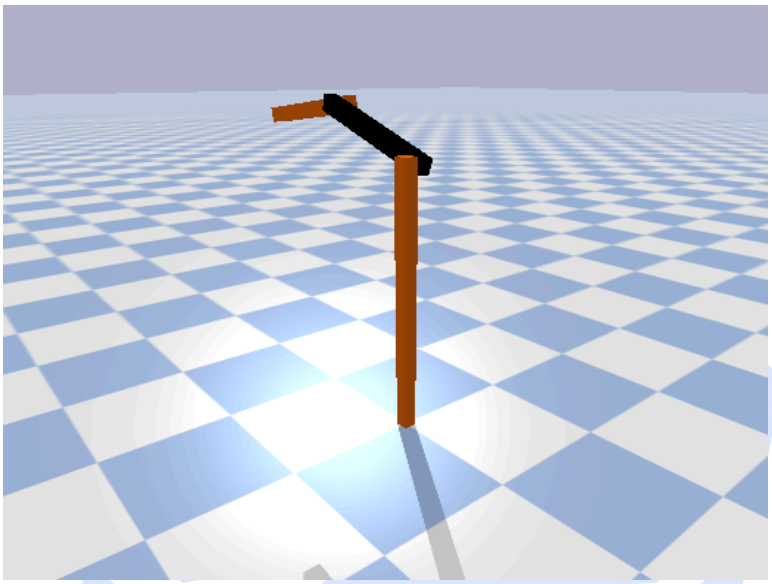
- For SolidWorks
  - Solidworks Complete Learning Tutorials
  - Solidworks Tutorials for beginners
- For Fusion360

    - Product Design Online Youtube Channel
    - Modelling Fusion360
    - Fusion360 tutorials for beginners
- Apart from these resources, you are free to refer to any relevant resource

Task : (Control & Simulation Task)

Task:

Your Task is to make the RR (Revolute-Revolute) robotic arm consisting of 3 links and 2 joints to draw basic shapes such as triangle,square and a circle.



Guidelines:

1. First, you have to start with learning what do you mean by a link, joint, forward kinematics, inverse kinematics, how a robot is being defined for simulation (URDF,SDF) , how the XML syntax is defined for each link and joint etc.
2. Get to know about PyBullet and how it differs from Gazebo or other simulation engines.
3. Use the URDF file linked here.

EVALUATION METRICS:

1. Completion of Task
2. The logic, shortness and well documentation of code.
3. Understanding of concepts and simulation workflow.

Resources:

1. Pybullet, Quick Start Guide [its like a treasure trove, explains all the API]

Spider R&D Inductions

Control System: Task 1

PROBLEM STATEMENT:

Your aim is to make the dot reach the destination position along the specified path.

Link to Script: python file

GUIDELINES:

1. The code for the window, maze and the GUI are given.

2. The initial and final destination points are also given.

3. You will be selecting the intermediate points by clicking on desired points in the corresponding window.

4. The dot must visit all the intermediate points and reach the final destination.

5. To implement this, you are required to develop 2 proportional controller functions - one for the movement along the x-axis and the other for the movement along the y axis.

6. So, basically the job of the controller is to make the dot reach all the turning points of the maze every time you click the mouse button.

EVALUATION METRICS:

1. Approach to solving the problem.

2. Neatness of the code and comments explaining steps

COMPONENTS REQUIRED:

1. Python 3. X

2. Pygame

RESOURCES:

1) Python
   a)
2) PID Control
   a) https://youtube.com/playlist?list=PLn8PRpmsu08pQBgjxYFXSsODEF3Jqmm-y
   b) https://www.tutorialspoint.com/control_systems/control_systems_controllers.htm
3) Pygame
a) https://www.javatpoint.com/pygame
b) https://realpython.com/pygame-a-primer/

Spider R&D Club

Embedded and Electronics Track

Electronics Practice Task - Tablet Bottling System

Task

A startup pharmaceutical company is gearing up to enter the market with a range of tablet medications. However, they face a significant challenge in efficiently and accurately bottling these tablets to meet regulatory standards and ensure product quality. As an electronics engineer design a tablet bottling system that helps the company in streamlining their bottling process by incorporating digital electronics into the tablet bottling system that have the precision needed for pharmaceutical-grade packaging.

Your Design should include these:

- A Keypad for entering the maximum number of tablets per bottle.
- The number entered should be converted to a suitable code for the working of other functional components.
- The converted code should be used to drive a seven segment display to display the maximum number of tablets.
- An IR sensor to detect each tablet that passes through it.
- The number of tablets passing through it should be counted.
- The number may be displayed on a seven segment display.
- Using the maximum number of tablets per bottle and the current number of tablets,draw an output that drives a motor.

Assume that tablets are circulating on a flat tray driven by a motor. The IR sensor is fixed close to the tray. Motor stops whenever the given numbers of tablets are filled in the bottle.

Guidelines:

1.Use only digital circuits.

2.Use Proteus for Simulation.

Resources:

Digital Circuits Tutorial (tutorialspoint.com) (Refer the subsequent topics of Combinational Circuits and Sequential Circuits)

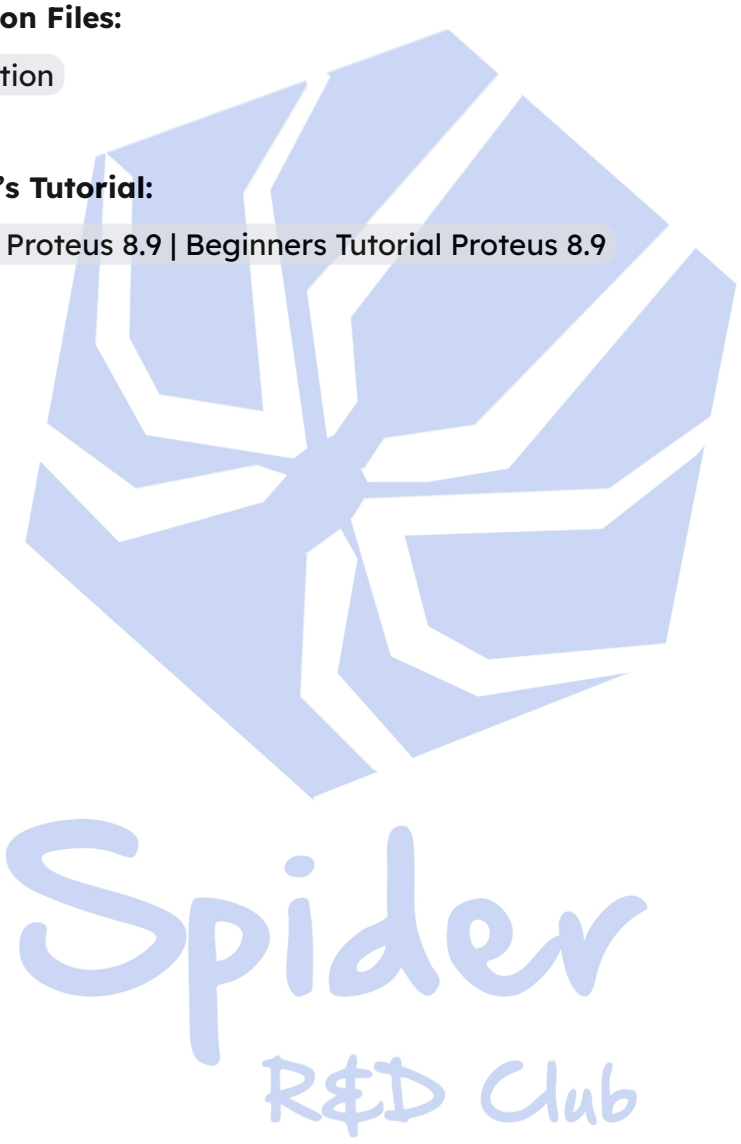
Installation Instructions:

Proteus Installation Files:

Proteus installation

Proteus Beginner's Tutorial:

Introduction to Proteus 8.9 | Beginners Tutorial Proteus 8.9



Embedded Practice Task - The Guardian

Task

A precious priceless stone is found and planned to be safeguarded in a museum. The stone is stable only in a narrow range of temperature from 23 degree Celsius to 35 degree Celsius. The museum also received a threat of robbery. As an embedded engineer help the museum authority to incorporate an optimum environment for the stone.

Your design should include these, adding further features is of your interest:

- **Temperature control:**

Use a temperature sensor to ensure that the optimum temperature is maintained. If optimum range is reached, indicate with a Temperature warning in LCD ,buzzer and RGB(red).

- **Secured Space:**

The stone is safeguarded inside a closed room in a secured space. The ultrasonic sensor indicates if someone is found in that space. If someone is found ,a warning is displayed in the LCD.

- **LCD Display:**

The live data of the warning inside the room is displayed in the LCD.

- **Alarm:**

There is a light which is incident on the stone every time. Think of the most appropriate sensor that detects when a person touches the stone. RGB(yellow) and a buzzer is triggered when this happens.

- **Password:**

A keypad is kept in the door, such that the door opens only when the right password is entered. Else if wrong password is entered RGB(blue) along with a buzzer is triggered.

- **Automated Shutter:**

Once the door is opened by officials, the door should close exactly at 3 seconds. A push button can be used to open the door. A servo can be used to symbolise this.

Note:

A single RGB is used and its colour is mentioned in respective operations.

Platform: TinkerCAD

Spider R&D Club

Guidelines:

1 Circuit must be designed in TinkerCAD.

2 Explore the available components and use them creatively.

3 You can use any of the Arduino libraries.

Resources:

Programming part of Arduino:

https://www.tutorialspoint.com/arduino/index.htm#:~:text=It%20consists%20of%20a%20circuit,code%20to%20the%20physical%20board.

Basics of Arduino (Includes Sensor interfacing):

https://youtube.com/playlist?list=PLGs0VKk2DiYw-L-RibttcvK-WBZm8WLEP&si=7FNqnmGRazlzVSZY

For LCD Display:

https://howtomechatronics.com/tutorials/arduino/lcd-tutorial

Spider
R&D Club

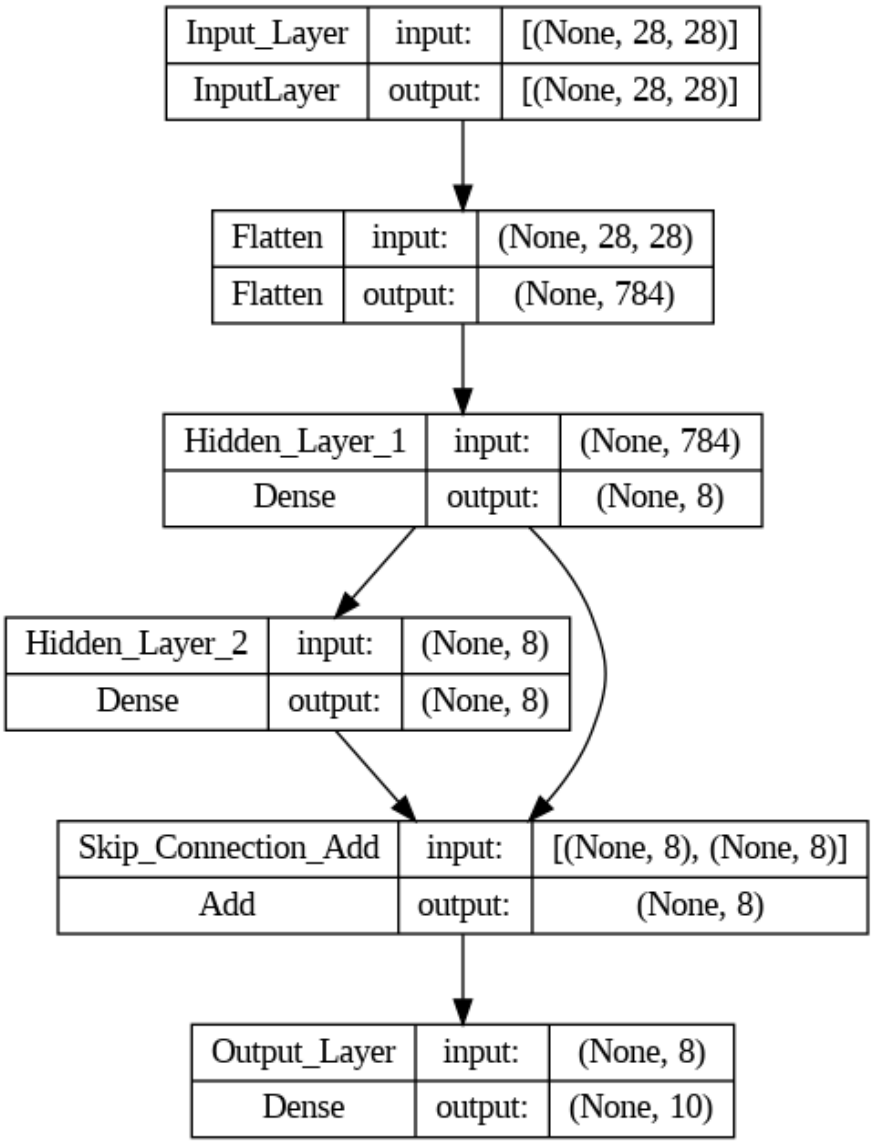
Machine Learning Track

Spider R&D Club

PART - 1

The task is to implement a neural network for the below-given architecture, **preferably** dynamic. You can make it dynamic by extending the micrograd course. You must implement both forward and backward passes and train it on the MNIST dataset given in the dataset here. You must also save your model weights (using Pickle) , and your notebook should contain a submission.csv consisting of predictions for the test set.

During training, you must print your train and validation(if you have split a portion from train set) losses and accuracy (prefer a plot for accuracy and loss vs epochs/iterations using matplotlib or any other library too).



Kaggle Dataset:
https://www.kaggle.com/datasets/frozenwolf/spider-ml-inductions-2023-task-1-part-1

Allowed Library: Numpy, Pickle, Matplotlib, CV2

Any other ML framework, such as PyTorch, TensorFlow, Keras, Theano, JAX and other libraries with autograd functionality, are **NOT** allowed.

Brownie points-
If the code is dynamic, refer to the micrograd course for it.

Submission:
Make a private Kaggle notebook for the following (mentioned in resources) dataset and save using "**Commit and run all".** Make sure the notebook is successfully run without errors before the given deadline. **ONLY** After the deadline you will be asked to set your notebook to public. The latest version that was run before the deadline will only be considered.

The dataset contains "trainingSet" which you can use to train the model, and the test_set, which contains the images for which you have to predict the labels. The predicted labels of the corresponding images need to be saved in submission.csv format, as shown in sample.csv.

PART - 2

Using **Pytorch** for this competition, train any **CUSTOM** model (**NO** pre-trained model) to train the given MNIST dataset in a different notebook. You must save your model weight, and your notebook should generate a submission.csv consisting of predictions for the test set. Your main goal is to find and train an efficient (high accuracy, low model size) neural network model and you are free to explore any architecture.

During training, you must print your train and validation losses and accuracy (prefer a plot using matplotlib or any other library too).

Allowed Library: Pytorch and any other libraries you used for task 1 are allowed, but any types of pre-trained models, weights, or external codes calling **pre-trained** models are **NOT** allowed.

Brownie Points:

- Explore Data Augmentation
- Checkout Class Imbalance
- Using Kaggle multiple GPU

Dataset Description:
The dataset contains trainingSet, which you must split into train and validation to train and validate the model, and the test_set, which includes the images for which you must predict the labels. The predicted labels of the corresponding images need to be saved in submission.csv format as shown in submission.csv (replace "?" with your prediction for the corresponding image).

Kaggle Competition Invite: (ONLY through this link you can join)
https://www.kaggle.com/t/aca31b03588149adb70d5c68fad5c0a7

Submission:

Make a **PRIVATE** Kaggle notebook for the following (mentioned in resources) competition and save using "**Commit and run all".** Make sure the notebook is successfully run without errors before the given deadline.

You can then go to the **OUTPUT** section of your notebook and **"SUBMIT"** the submission.csv file generated. **ONLY** After the deadline you will be asked to set your notebook to public. You can do multiple submissions while choosing which to consider for the final submission. Your aim is to get the highest leaderboard score while having a small model size.

Check this for a more detailed explanation of how to make a Kaggle submission in competition.

(**Tip***: First try to hit the highest accuracy and then start decreasing your model size*)

Once you have submitted your notebook and submission.csv (any number of attempts) files in Kaggle to get your best score **BEFORE** the deadline, you must submit the final 2 URLs for **BOTH part-1** and **part-2** in the portal. **AFTER** the **deadline**, set all of your notebooks to the **PUBLIC**.

Evaluation Criteria:

Part -1: Your idea/Solution and your code implementation.

Part-2: How much you have experimented, Accuracy, Code Implementation, Efficient experimentation

Starter Notebooks:
How to save your predictions:
https://www.kaggle.com/code/frozenwolf/how-to-correctly-save-your-prediction
How to Submit:
https://www.kaggle.com/code/frozenwolf/how-to-submit-your-notebook

Resources:

1. https://www.coursera.org/specializations/deep-learning
2. http://www.cse.iitm.ac.in/~miteshk/CS6910.html
3. Micrograd course:
   The spelled-out intro to neural networks and backpropagation: buil…
4. Pytorch:
   Learn PyTorch for deep learning in a day. Literally.

   https://youtube.com/playlist?list=PLhhyoLH6IjfxeoooqP9rhU3HJIAVAJ3Vz

5. Intro to Kaggle Competition:
   https://www.kaggle.com/code/alexisbcook/getting-started-with-kaggle-competitions