RSA Mini-Hackathon - Line Following Tank

1 Setup

1.1 Hardware

The tank is controlled through one Arduino Mega with a motor shield.

Each component in the tank are listed in Table 1.1 with its corresponding port name and the Arduino I/O pin.

Component	**Port**	**Pin**	**Description**
LCHB-100 (motor driver)	1REV	7	Applies torque to the left motor in reverse at a speed depending on the analog value.
	1EN	24	Enables the left motor. Pulling this down to ground stops the motor abruptly.
	1FWD	6	Applies torque to the left motor in forward direction at a speed depending on the analog value.
	2REV	3	Applies torque to the right motor in reverse at a speed depending on the analog value.
	2EN	25	Enables the right motor. Pulling this down to ground stops the motor abruptly.
	2FWD	2	Applies torque to the right motor in forward direction at a speed depending on the analog value.
HC-05 (Bluetooth module)	TX	18	Sends serial binary data to the connected device wirelessly.
	RX	19	Obtains serial binary data from the connected device wirelessly.
HC-SR04 (ultrasonic distance sensor)	Trigger	23	Allows this sensor to send out an ultrasound normal to its front.
	Echo	22	Obtains the ultrasound that was sent by the trigger pin.
Yellow LED	-	13	Let there be light!
TCRT500 (reflective infrared sensor)	-	8	Left optical light sensor. Detects if the light that this sensor sent out is reflected back.
	-	9	Right optical light sensor. Detects if the light that this sensor sent out is reflected back.

1.2 Software

This is the list of software you need for the hackathon:

- Arduino IDE
- Python 3.9 and above
- Visual Studio Code

1.3 Arduino IDE

For this hackathon, you would need to download and install Integrated Development Environment (IDE) for Arduino. This software lets you rewrite and improve the code that controls the Arduino of your tank. The download link is as follows: `https://www.arduino.cc/en/software`. Follow the instruction about installing Arduino on the site, if required.

1.4 Visual Studio Code

If you do not already have this installed, installing it is trivial. Download the installer from the following link: `https://code.visualstudio.com/Download` and install it. For those who don't know what it is, it is a program that is lets you write codes and develop software with ease. It could be useful for the rest of your studies as it is quite versatile!

1.5 Python

Python is needed for this hackathon. These instructions are windows specific.

1.5.1 Check if Python is already installed

To check if you already have python installed open a command prompt(cmd) and write the command:

```
python --version
```

You should get something like Python 3.9 or Python 3.10.

If you get an error go to this website `https://www.python.org/downloads/windows/` and download the latest x64 installer.

1.5.2 Create a virtual environment with venv

Don't get scared by the word virtual environment, it just means a separate place to install your libraries so you keep each project separate. In a command prompt, go to *"my Documents"* using the following command:

```
cd Documents
```

Then go to the folder you just cloned from Gitlab with:

```
cd RSATankHackathon
```

Once in there, use the command:

```
python -m venv rsa-tank-env
```

A virtual environment called `rsa-tank-env` has been created. Now, issue the command:

```
.\tank-env\Scripts\activate.
```

In your cmd you should now see the name **rsa-tank-env** all the way to the left. Now, issue the command:

```
pip install -r requirements.txt
```

All dependencies needed for the hackathon will then be installed. The setup is now complete!

1.6 Additional tool

If you're going to use VSCode, install the **Python Extension Pack**. It will make your life easier.

2 Usage and examples

Now, you have set up everything you need to participate in the hackathon. This is an explanation of different practical matters such as:

- Connect the Arduino to your computer.
- Pair your computer to the tank via Bluetooth.
- Use remote control via Bluetooth.

2.1 Using an Arduino

This is very simple. Follow these steps:

- Open the Arduino IDE.
- Get a blue USB-A cable.
- Connect that cable to your laptop and to the Arduino.
- Now in the Arduino IDE, go to **tools**.
- Look for the **board** menu and select **Arduino Mega ADK**.
- Now in the same menu find **port** and select the port of the Arduino connected to your laptop. You should see something like *COM1* there. Select it.
- Now you can change the code and upload it to the Arduino!

2.2 Bluetooth remote control

Once the Arduino is connected to your laptop, a serial port opens that can be used to send and receive information. Using Bluetooth is similar to this, but windows must first get configured to let a Bluetooth device open a serial connection.

2.2.1 Pair your tank with your laptop

- Open your Bluetooth menu.
- Click on **add Bluetooth or other device**.
- Click on **Bluetooth**.
- Wait until you see **Robot (YOUR TANK NUMBER)**.
- Pair to it using the code 1234.
- Once paired find **more Bluetooth settings** and find the tab **COM ports**
- Click on **add**. There click on **outgoing** for the connection type and **HC-05** or **HC-06** for the device.
- Now, remember your COM port because it's needed for the next step.

2.2.2 Use the python script for remote control

Now, this part should be easy assuming the python virtual environment from section 2.7 is properly set up.

- Open Visual Studio Code.
- Open the folder where you cloned the code to within VSCode.
- At the bottom left corner where you should see your python interpreter(something like *python 3.9 64-bit*). Click on it and change it to **rsa-tank-env**.
- Now open the Python script inside that folder in VSCode.
- At the line

  ```
  arduino = serial.Serial('COM22', 57600)
  ```

  enter the COM number your Bluetooth serial port is using and make sure your laptop and the tank are paired.
- Right-click on it and click **run python file in terminal**
- Enjoy remote control using the keyboard!

3 Hackathon Rules

3.1 Game Field

The following graphic shows an approximation of the game field with the different areas.

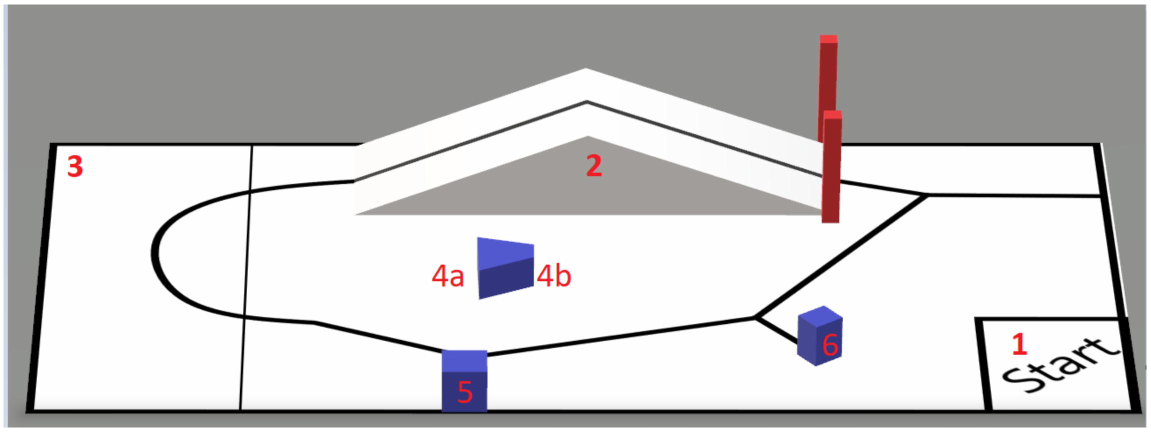

Figure 1: Game field overview.

Some areas are marked with numbers to identify them. Take a look at the actual game field to understand which areas are referred to.

1. Start/finish area
2. Ramp
3. Chip placement area
4. Wide and narrow gap obstacle
5. Large box
6. Small box

The game field also consists of small red unmovable tubes, which were not in the graphic for simplicity reasons. Please be careful to avoid colliding with them.

Do not hesitate to ask board members for more information.

3.2 Game Objects, Positioning

Every team will have access to two **(2) table tennis balls and two (2) RSA chips for each run**.

The **table tennis balls**, if used, must be placed within the start area. They can be either on top of the robot or even on the floor if the team decides to. The **RSA chips** must be placed in the **Chip Placement Area** as shown in the Figure 1. That area is the small rectangle within the game field walls and the narrow black line. They can be placed in **any way** the team wants, as long as they are within the corresponding area.

3.3 Robot missions

In this section, the robot missions will be presented and explained. Then, their corresponding points will be discussed in the following chapter.

3.3.1 Finish within the starting area

This challenge is simply to park the tank at the end within the black square area where it started. Please see Figure 1 number 1.

- The robot will get half points if it touches the black outline, but the rest of the tank is within the square.
- The robot will get 1/4 of the points if it just touches the black outline and it is partially outside the start area.

For the finishing points to count, the robot should get at least 5 points from another challenge.

3.3.2 Line following

The main software task is the very famous line following! Each robot is equipped with two light sensors at the front of the robot, and you can use one or both of them to make your robot follow the line. The robot manual and the code already provided to contains a lot of information on how to use them, please read them carefully.

In order for the robot to be following the line and be able to receive the points, the robot must cover at all times any part of the black line.

Also, in order to be able to get points from these challenges, the robot must move autonomously.

The following graphic shows that the game field is split into multiple sections for the line following challenge. Each section is graded independently, and more importantly, **it doesn't matter the order, or the orientation of the robot, as long as the robot follows the line one way or the other**.

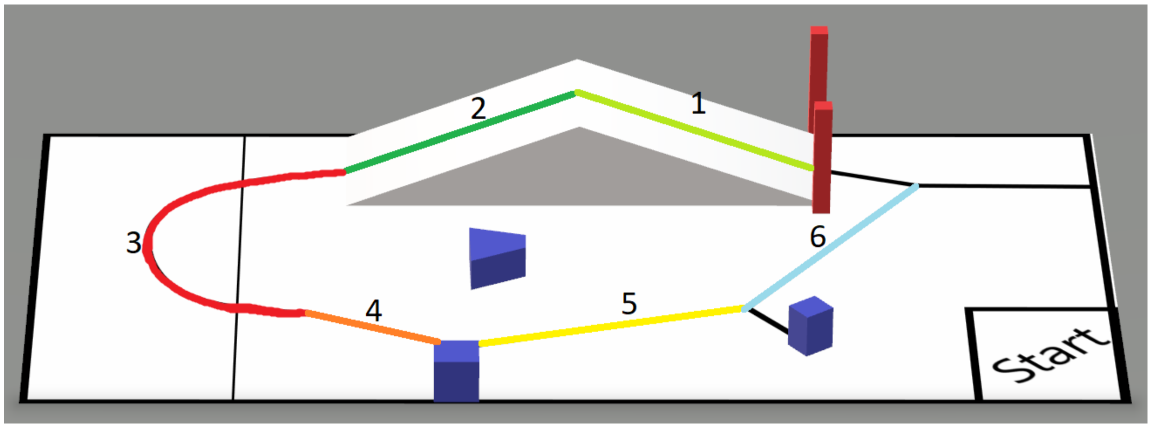

Figure 2: Line following sections colour and numbered.

3.3.3 Hill climbing (only mechanical)

This challenge is about moving your tank upwards and downwards the ramp while manually controlling it via Bluetooth. In the robot manual, it is explained how you can set up wireless communication.

There will be separate points for going up and down the ramp.

3.3.4 Ping-pong ball dropping

As mentioned before, each time can have up to two ping pong balls each round. The challenge is to drop these two ping pong balls inside the two boxes listed as numbers 5 and 6 in the Game Field Figure 1.

You can earn points for only one ball per box. For example, if two balls are dropped in the small box, only the first ball will receive points, and the second ball will be ignored.

Double points for this part will be earned if this challenge is completed autonomously.

3.3.5 Chip pushing

The goal of this challenge is to push the two RSA chips in the object listed as number 4 in Figure 1. There are two openings, one wider than the other, with different points corresponding to each opening. The chip must pass the black line within the object to earn the points.

Double points for this part will be earned if this challenge is completed autonomously.

3.4 General Rules

Before moving on with the scoring there are few general rules for the whole hackathon.

1. You can use the course/game field to test your robot, but please remember that other teams may want to try their robot. Please do not spend much time on a single test-run if other groups are waiting. We do not want to set a time limit, so please be respectful to other teams.

2. You are allowed to reposition the sensors as you want, as longg as you don't permanently glue them to the robot skeleton.

3. The robot must not damage the game field in any way. Field points will be deducted from their final score if teams damage the game.

4. There will be some materials in the **Workshop** that you can use to complete the mechanical challenges.

5. Throughout the hackathon, you can have up to 5 runs. Each run has a maximum duration of 3 minutes; the run ends at three minutes unless someone from the team touches the robot during the run, then the robot must stop at that specific point. There will be three graded sessions throughout the event when your team can have their run be graded.

   - Lunch-run → 12:30
   - Afternoon-run → 14:30
   - Final-run → 18:00

   You need to have at least one run during **Final-run**, but feel free to spread your runs as you please. To register for graded runs, **ONE** team member needs to fill in the Run-Sheet at least 10 minutes before each run.

6. You are not allowed to modify the code or the robot between runs in the same session.

7. You are allowed to place the robot to start outside the designated area, but this will cost your team 5 penalty points.

8. Your final score will be the highest score among your runs. If there is a tie, the winning team will be the team with the faster run.

3.5 Scoring

3.5.1 Scoring sheet

	Challenge	**Max Points**	**Completed (Y/N)**	**Penalty**	**Total**
1	Finish within the starting area	20			
2	Line following section 1	20			
3	Line following section 2	10			
4	Line following section 3	20			
5	Line following section 4	10			
6	Line following section 5	10			
7	Line following section 6	10			
8	Hill climb uphill	10			
9	Hill climb downhill	5			
10	Ping pong ball large box	15			
	Bonus if autonomous	15			
11	Ping pong ball small box	20			
	Bonus if autonomous	20			
12	Chip pushed in the wide gap	10			
	Bonus if autonomous	10			
13	Chip pushed in the narrow gap	10			
	Bonus if autonomous	20			
	Grand total				