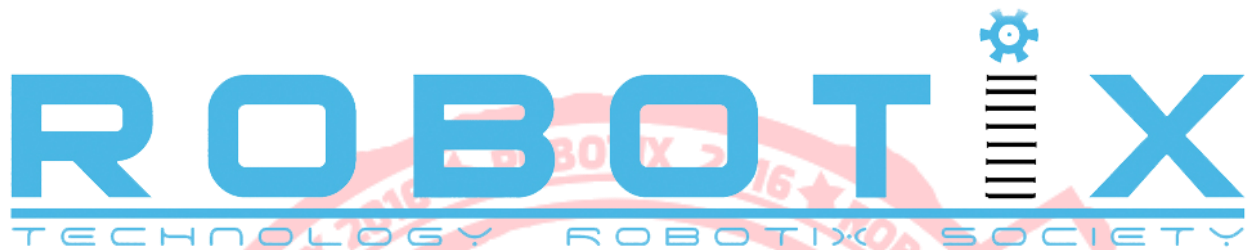


INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
KSHITIJ 2016

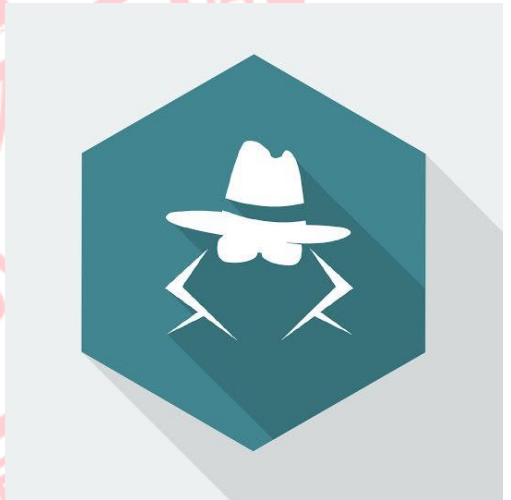


Sherlock

Autonomous Robotics

USP:

IR Hex Decoding and Heading Following



Introduction:

Sherlock is stuck in a featureless desert wasteland with nothing but a compass in his pocket. He must find his way out using the clues scattered in the desert.

Problem Statement:

Build an autonomous robot that can follow compass headings to go from start to finish using IR receiver to receive arena and waypoint information, while optimizing its path.

Description:

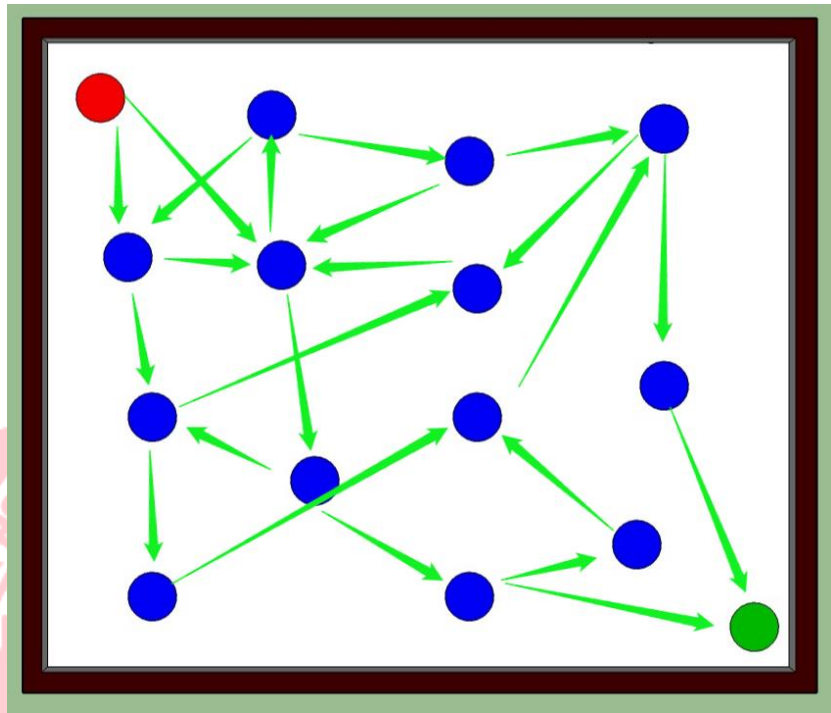
- The arena consists of a **start**, a **finish point** and other intermediate **waypoints**. The start, finish and waypoints are collectively referred here as **POI (points of interest)**.
- Each POIs is identified by its **unique ID**.
- These POIs have a **circular area of 5 cm radius** with IR transmitters on the centre. The IR transmitter transmits ID of the respective POI and other information as stated below.
- The information is transmitted through messages using **NEC IR protocol**. For more information, refer the [Tutorial](#).
- A message starts with **400** as **start code** and **ends** with **500** as **end code**. **450** is the code used as **separator** in between the messages.
- A POI may transmit multiple such messages (max 9). Then each message will be tagged with a two digit message_tag.
- The first digit denotes the present message id, while the second digit denotes the total number of messages to be transmitted from that POI.
- Heading and magnetic heading as mentioned henceforth, refer to the direction **with respect to the magnetic north direction of the Earth at that position**.
- So if heading to a point is 90 degrees, it points towards East.

Arena:

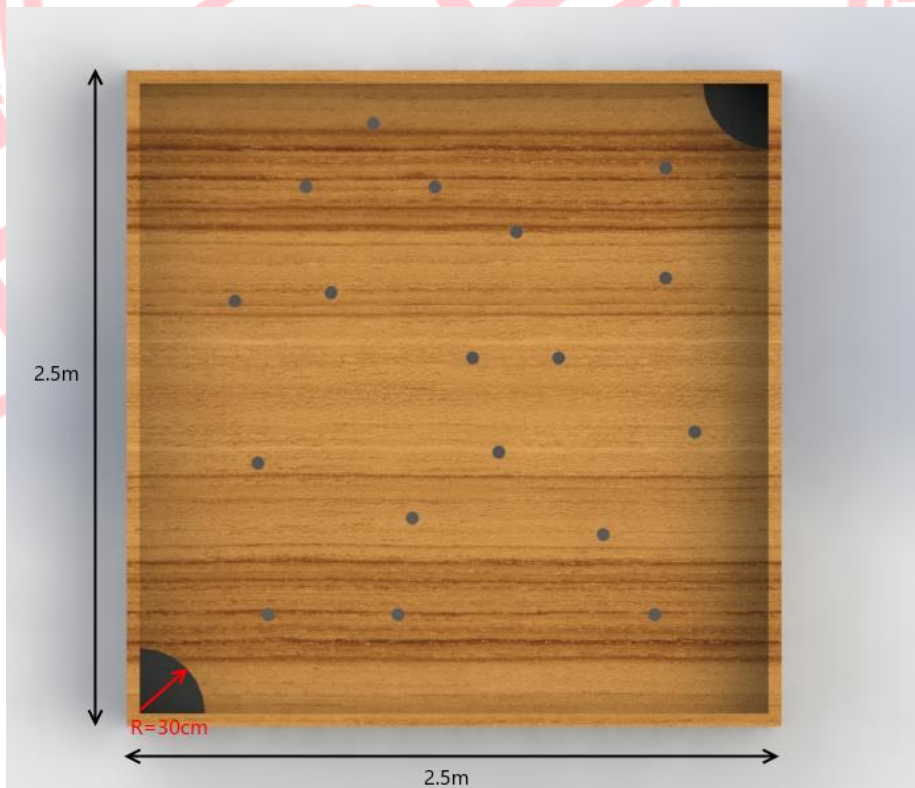
Depictive Arena

- Legend:
 - **Red POI:** Start point
 - **Blue POI:** Waypoints
 - **Green POI:** Finish point
 - **Green Lines:** Valid edges

Round 2



Actual Arena



Task:

Round 1

- Each waypoint will provide heading to a **single POI**. For example, waypoint 2 transmits the info for waypoint 4 in format as follows:

start message_tag own_id [id heading cost]s stop

400 11 2 4 260 20 500

Here 400 is the start code, 11 is message_tag (1st of the 1 total messages), 2 is the ID of the current waypoint, and it is connected to waypoint 4 at headings of 260 degrees via an edge of costs 20.

- When the participant's run starts, the bot is placed on start point. The start point will provide the following information:

- A1: **Finish POI for that particular run**. Eg:

start msg_tag own_id finish_id stop

400 12 0 21 500

Here 400 is the start code, 12 is the message ID (1st of the 2 total messages), 0 is the ID of the current POI and 21 is the ID of the finish POI, 500 at the end being the stop code.

- A2: **Heading for only the first possible waypoint after itself**. Eg:

start msg_tag own_id [id heading cost]s stop

400 22 0 2 260 20 500

Here 400 is the start bit, 22 is the message_id (2nd of the 2 total messages), 0 is the ID of the current POI and the next waypoint is the POI with the ID 2 at a heading of 260 degrees and cost 20, and finally 500 is the stop code.

- **All the edge costs are equal to 20.**
- The bot has to reach the final POI, which is 21 in the above example.
- The bot has to follow only the **valid edges**. If waypoint 2 directs the bot to waypoint 3, then it has to head towards waypoint 3 only, otherwise there would be a penalty.
- There will be no explicit markings on the arena to indicate valid edges.
- If the bot gets lost in the arena and stumbles upon a POI, then it has to plan a way to continue its pre planned path going through valid edges.

Round 2

- Each POI will provide heading to a **single or multiple different other POIs**. For example, waypoint 3 transmits that it's connected to waypoint 2,6 and 5 in format as follows:

start message_tag own_id [id heading cost]s stop

400 11 3 2 260 10 6 180 45 5 300 39 500

- Here 400 is the start code, 11 is message_tag (1st of the 1 total messages), 3 is the ID of the current POI, and it is connected to POIs 2, 6 and 5 at headings of 260 degrees, 180 degrees and 300 degrees via edges of costs 10, 45 and 39 respectively.
- When the participant's run starts, the bot is placed on start point. The start point will provide the following information:

- A1: **POIs each waypoint directs to.** Eg:

start msg_tag own_id [waypoint [waypoint_it_directs to]s separator]s stop
400 13 0 2 3 6 5 450 3 5 2 7 450 500

Here 400 is the start code, 13 is the message_id (1st of the 3 total messages), 0 is the ID of the current POI. Waypoint 2 is connected to waypoints 3, 6 and 5. Waypoint 3 is connected to waypoints 5, 2 and 7, and finally 500 is the stop code. The 450s as mentioned are separators.

- A2: **Finish point for that particular run.** Eg:

start msg_tag own_id finish_id stop
400 23 0 21 500

Here, 400 is the start code, 23 is the message_id (2nd of the 3 total messages), 0 is ID of the current POI and 21 is the ID of the finish POI, 500 at the end being the stop code.

- A3: **Heading for only the first possible waypoint after itself.** Eg:

start msg_tag own_id [id heading cost]s stop
400 33 0 2 260 25 500

Here 400 is the start code, 33 is the message_id (3rd of the 3 total messages), 0 is the ID of the current POI and the next POI is the POI with ID 2 at heading of 260 degrees and cost 25, with 500 being the stop code.

- So the bot gets to know the **entire connected graph** on the start POI.
- The bot has to plan its route passing through the **minimum number of POIs** from the start POI and minimising the cost of travel along its way to reach the predefined finish POI while, traversing through **only the valid edges**.
- There will be no explicit markings on the arena to indicate valid edges.
- The edge cost between two POIs is a **biquadratic function of the absolute value of their difference in their IDs modulo 5**. The cost between POIs with ids m and n is defined thus as:

$$\text{cost}(m, n) = ax^4 + bx^3 + cx^2 + dx + e$$

$$\text{where } x = |m-n| \% 5$$

- Where % is the **modulo operator**, denoting the remainder when absolute value of m - n is divided by 5, which can be 0, 1, 2, 3 or 4.
 - Thus cost of an edge between POI 2 and 16 is $256a + 64b + 16c + 4d + e$ as $(16-2)\%5=4$.

- All the constants **a, b, c, d, e** are all non-negative integers.
- The constants a, b, c, d, e are unknown and the bot has to find them itself using techniques of simultaneous equations. The robot knows the edge IDs and can solve five equations after getting sufficient information.
- Please note that this biquadratic edge cost term comes into play only in round 2, round 1 has all edge costs equal to 20.
- If the bot gets lost in the arena and comes back to a POI, then it has to plan a way to continue its pre planned path going through valid edges.

Specifications:

Round 1

- Each waypoint will send the bot to **only one other waypoint**.
- The bot must simply follow the directions to the finish POI.
- The round will be of **3 minutes duration**, after which the run would be stopped immediately and score would be calculated.
- If finish point is reached then the bot has to **flash an LED**.

Round 2

- Waypoints will be transmitting heading and cost information for **single or multiple other waypoints**.
- Start POI will transmit the heading and cost for its immediate neighbour POIs and providing referencing information for the entire arena, i.e. which waypoints each waypoint on the arena provides the headings for.
- Robot has to plan its best **optimized path** with the information being given from the start POI as mentioned in Task above.
- Once the bot reaches the finish point it must **flash an LED**.
- The round will be of **5 minutes** duration, after which the run would be stopped immediately and score would be calculated.

Rules and Specifications:

General Rules

- The robot should be **completely autonomous**.
- All arena dimensions have a tolerance of **10%**.
- Each team can consist of a **maximum of 4 members**.
- Each team should have unique participants i.e. no two teams can have even a single participant common.

- The team members can be from different institutes or colleges.
- The right spirit of participation is expected from the participants.
- **The decision of the Team ROBOTIX will be final and binding.**

Technical Rules

- A standard **220 volt AC supply** will be provided by Team Robotix in the arena.
- All circuitry and sensory equipment should be placed on the robot adhering to the ROBOT SPECIFICATIONS.
- Participants will have to bring their own programmers, cables and softwares. No programmers will be supplied.
- **Hard coding is NOT allowed.** Hard coding is defined [here](#).
- The robot can be powered on-board as well as off-board.
- **No kind of external control will be allowed.**

Robot Specifications

- Each robot can have maximum dimension of **25x25x20 cm³ (LxBxH)** respectively.
- No part/mechanism of/on the robot should exceed the given dimensions before the commencement of the event run. The robots can exceed their respective dimensions once the event commences.
- The autonomous robots should be on-board processing robots, i.e., the robots cannot be controlled by a remotely kept computer.
- [LEGO kits](#) or its spare parts or pre-made mechanical parts are not allowed.
- Ready-made gearboxes, sensors, development boards can be used but no other part of the robot should contain any ready-made components. Simple car bases with no extra features may be used.
- The bots should not harm the **Sherlock** event arena in any way. If it does so, a penalty will be imposed on the team. The magnitude of the penalty will be decided by **Team ROBOTIX**.
- Processors of **less than 16-bits are allowed**. ARM processors are not allowed.

Event Rules:

Robot:

- The robot should be completely autonomous.
- The robot needs to traverse a **2.5m x 2.5m** arena for both rounds
- The arena has multiple number of IR LEDs (POIs) that transmit information according the NEC IR Protocol.

POIs:

- The POIs are small holes in the arena that have an IR LED inserted into each of them. The range is such that the **TSOP1738 IR Receiver** can detect a POI from a distance of about 2 cm from the POI.
- The POIs would transmit binary data at **38KHz frequency in strictly NEC IR format**. The format as detailed in the [tutorial on the website](#).

Magnetic Heading:

- The magnetic heading would strictly be calculated with respect to north pole of earth wrt IIT Kharagpur campus. Although, the code doesn't require calibration as long as it is written anywhere in India/neighbours
- We have used [Adafruit HMC5883L library](#) for calibration. Usage of other compass modules or other libraries is not recommended and any discrepancies thus caused would not be the responsibility of Team Robotix.
- The Magnetic Heading would be given in **degrees** from 0 to 359 and **not in radians**.

Restarts/Timeouts:

- In **Round 1**, the participant can take a maximum of **1 timeout of 2 minutes**. In **Round 2**, **2 such timeouts** may be taken. Penalty will be imposed for each timeout and the robot will start from the last POI crossed.
- In Round 1, the participant's robot can have a maximum of **2 restarts**. In Round 2, the participant's robot can take **2 such restarts**. A penalty will be imposed on the team for every restart that they take. In case of a restart the participant's robots will be placed on the start POI. The timer will be **reset to zero** and the run will **start afresh** with the addition of the penalty for the restart.
- Restarts will only be awarded to the participant in case of a technical failure of the bot.
- If the robot goes outside the arena, then the team has to take a restart or call off their run.

Scoring

Round 1

Base Score: 1000

Rewards:

- Reaching the Finish POI: **500 (F)**

- Reaching each valid POI: **100 (P)**

Penalties:

- Traversing an invalid edge once: **-200 (X)**
- Timeout: **-250 (T)**
- Restart: **-400 (R)**

Scoring Formula:

- The final score is given by:
 $1000 + 500*(F) + 100*(P) - 200*(X) - 250*(T) - 400*(R)$

Round 2

Base Score: 2000

Rewards:

- Reach Finish POI: **1000 (F)**

Penalties:

- Traversing an invalid edge once: **-200 (X)**
- Traversing each POI: **-A (P)**
- Sum of edge costs incurred in reaching the Finish POI: **-B (TC)**
- Timeout: **-500 (T)**
- Restart: **-800 (R)**

The constants A and B will be declared during the fest.

Scoring Formula:

- The final score is given by:
 $2000 + 1000*(F) - 200*(X) - A*(P) - B*(TC) - 500*(T) - 800*(R)$
- The team with the maximum score wins.
- If two participants manage to get the same score, then the participant with lowest time taken is declared the winner.

Tutorial & Resources

Visit www.robotix.in/event/sherlock to check out the latest Event Updates.

Read our [Tutorial](#) for Sherlock.

Join the [Event Facebook Group](#) for latest updates and doubt sessions.

Contact

For queries, contact our Event Heads:

Aditya Narayan

Riya Bubna

Mratunjay Gupta

aditya.narayan@robotix.in

riya@robotix.in

mratunjay@robotix.in

+91-8609283200

+91-9007388802

+91-8609266778

