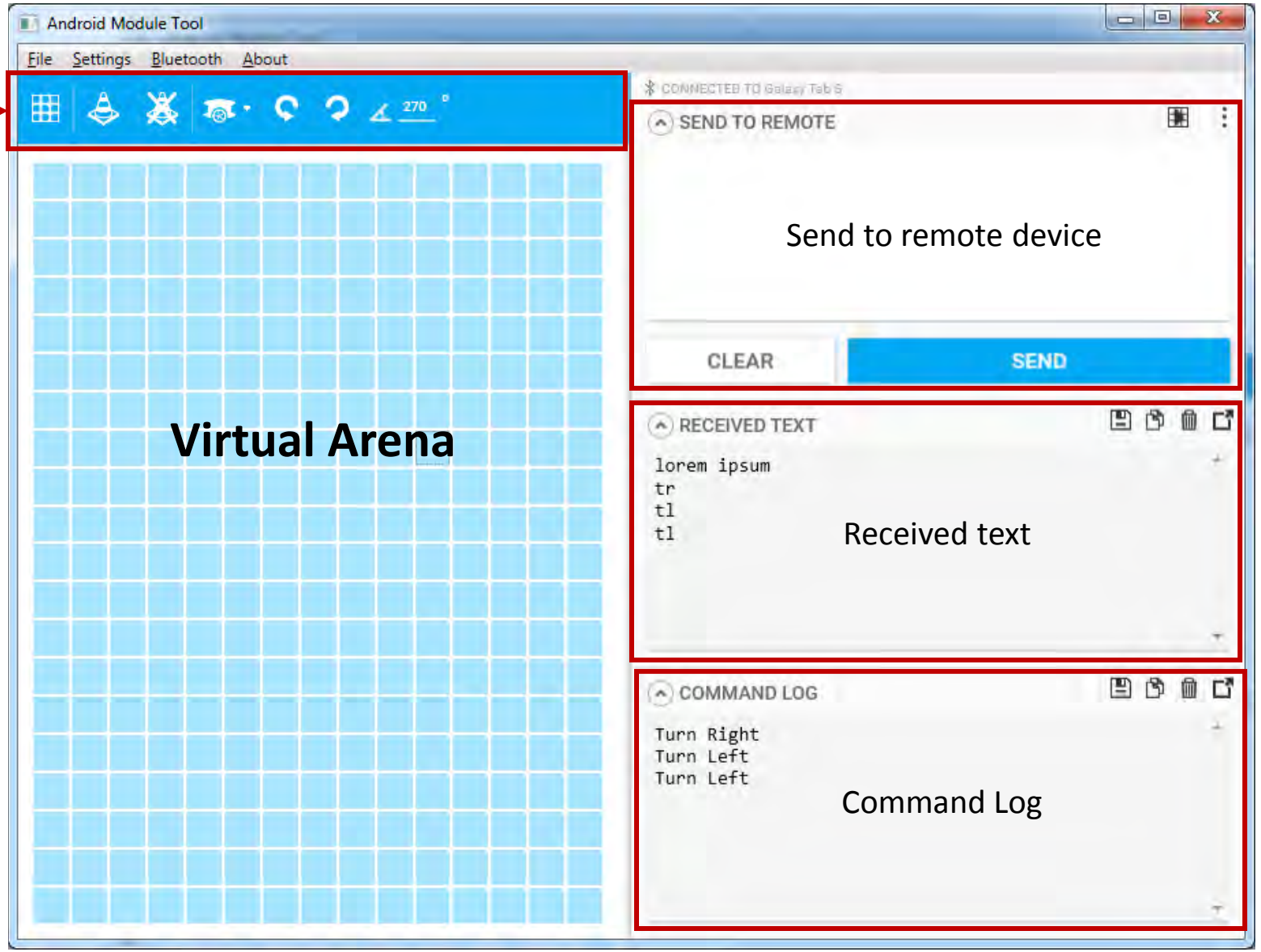
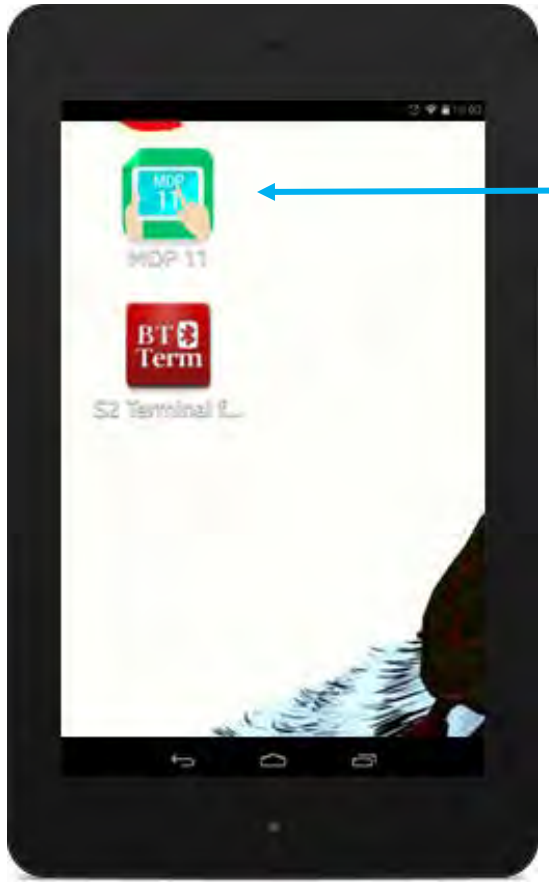
Main Interface Components

Virtual Arena tools



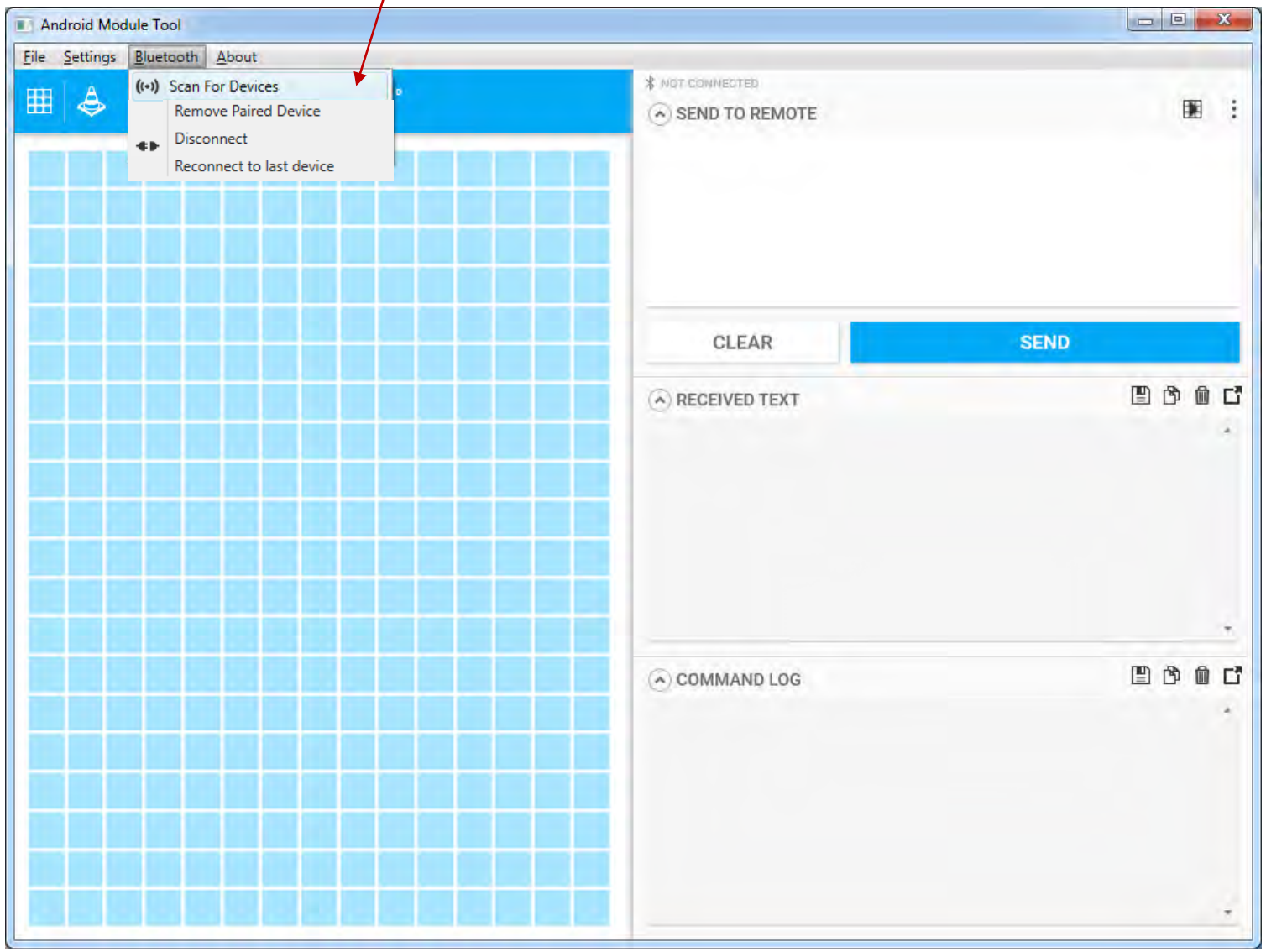
Connect to Bluetooth Device (As client)

Open the app on your device first.
Alternatively, you can open the “S2 terminal for Bluetooth” Bluetooth terminal emulator program that can be downloaded from the Play Store

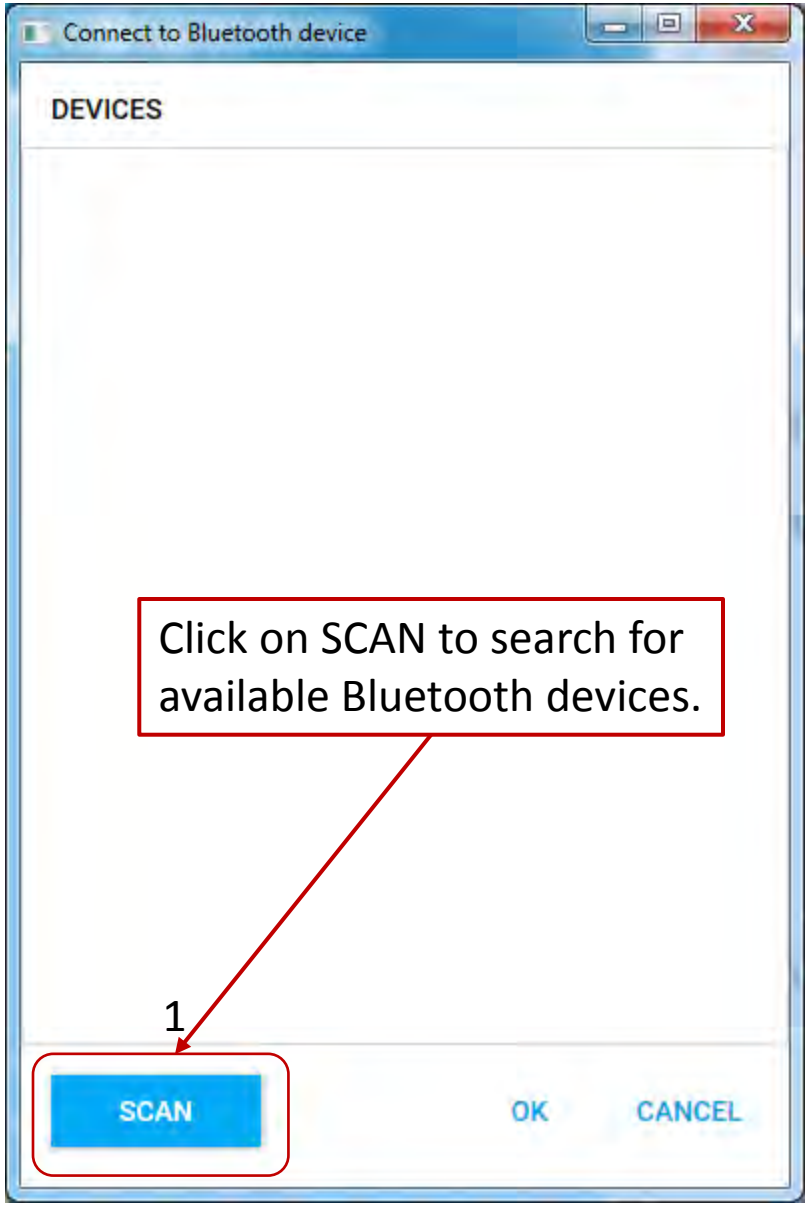


Select the MDP app before connecting the AMD tool app on PC

Connect to Bluetooth Device (As client)

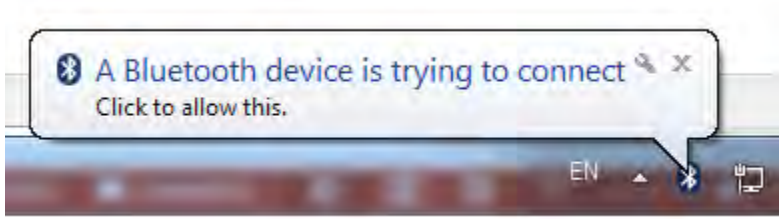


Connect to Bluetooth Device (As client)

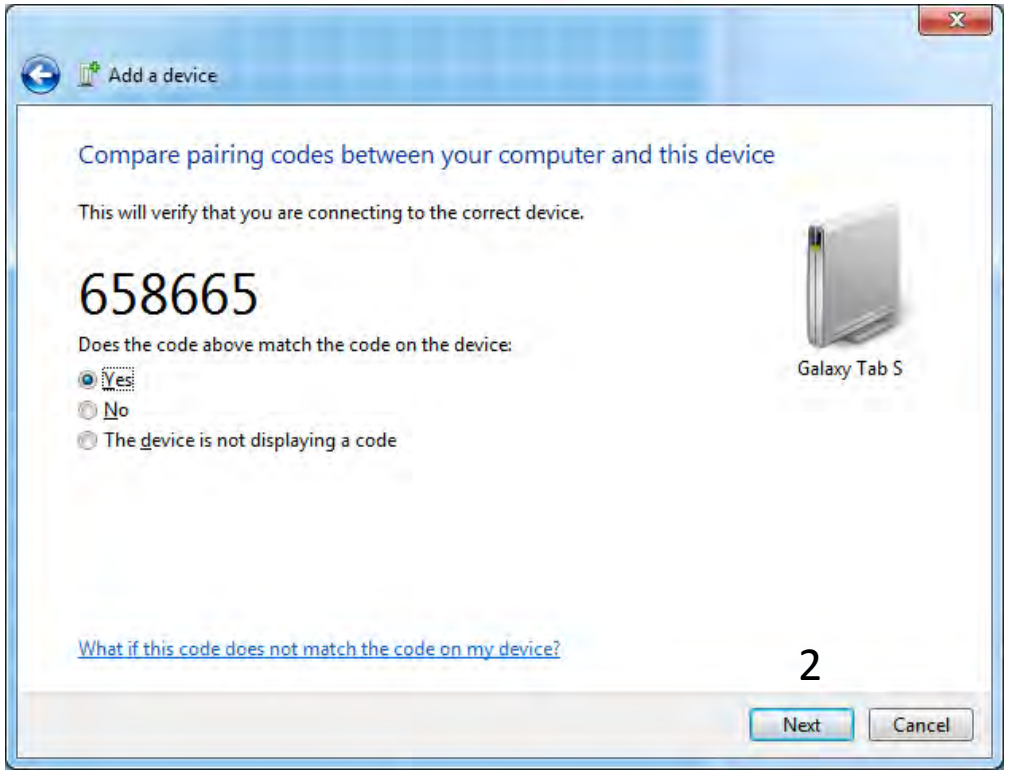


Connect to Bluetooth Device (As client)

If connecting to the device for the first time.
Compare code on the device with this code and click “Next”.
Also allow the pairing on the device itself.



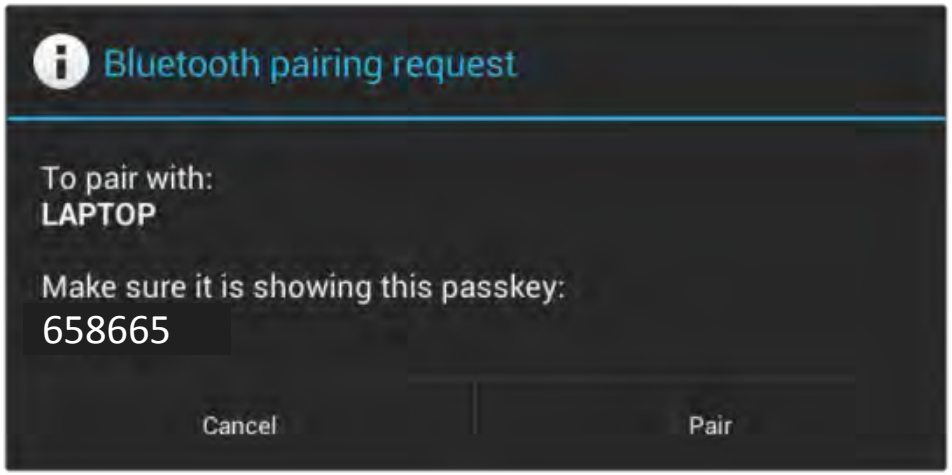
1



2

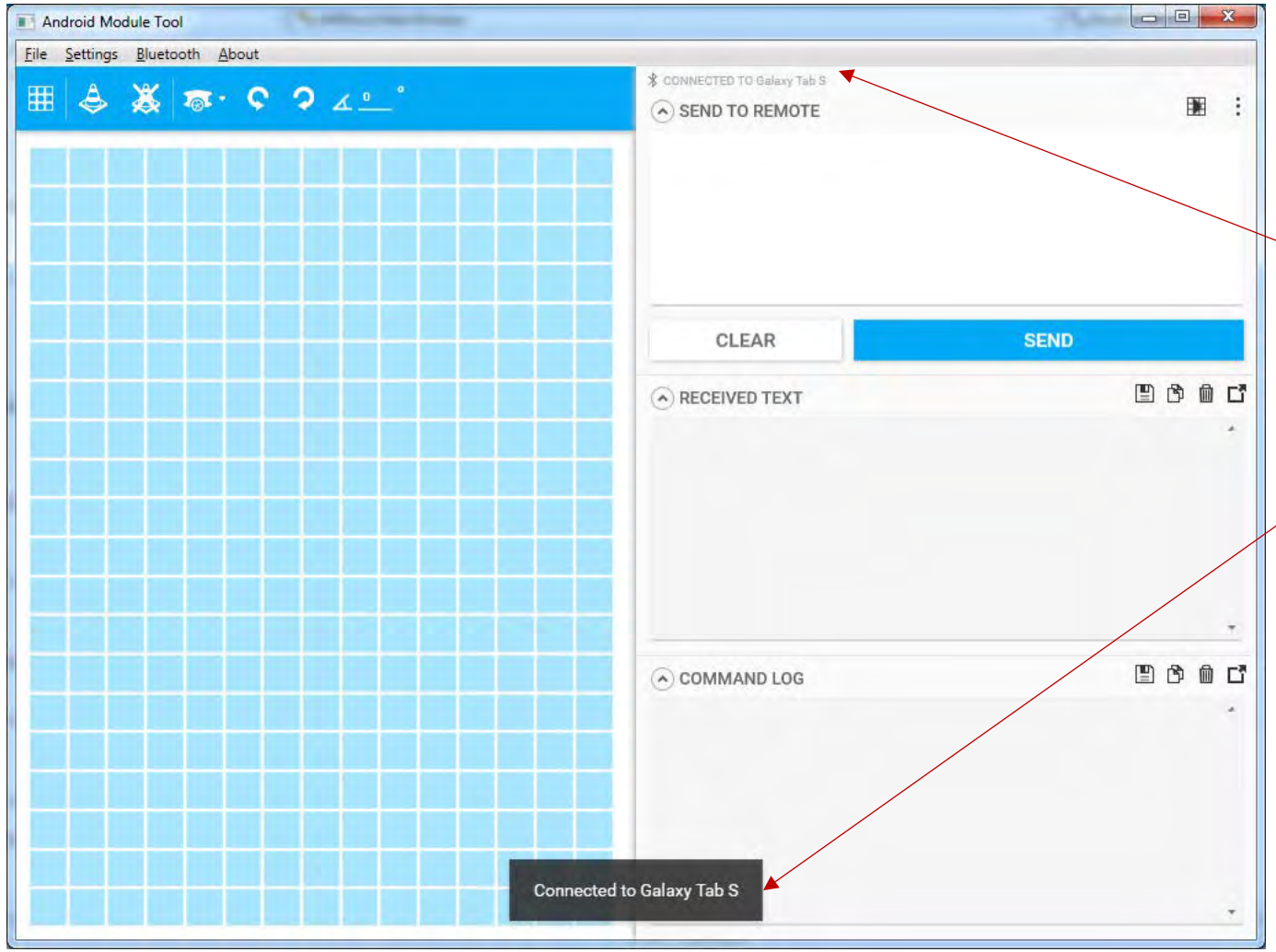
(Or equivalent message)

(Or equivalent message)



2

Connect to Bluetooth Device (As client)



Indication of successful connection to device

Virtual Arena

Place obstacle

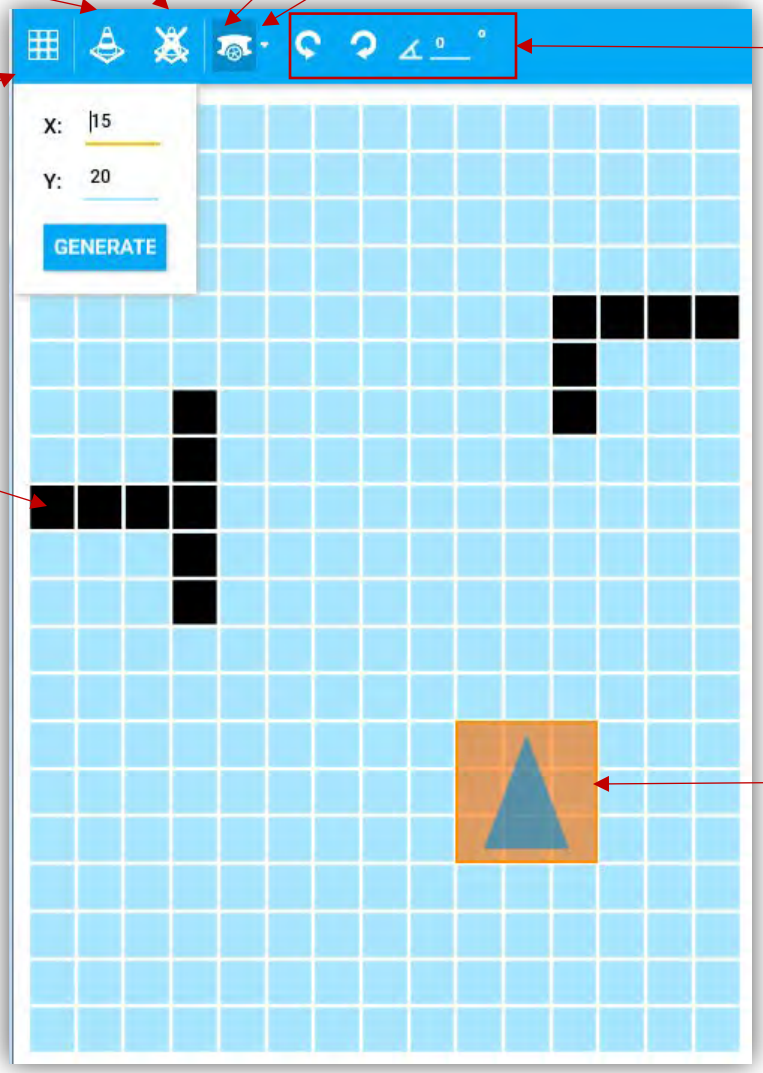
Remove obstacle

Place robot

Change robot size

Generate Arena

Obstacle

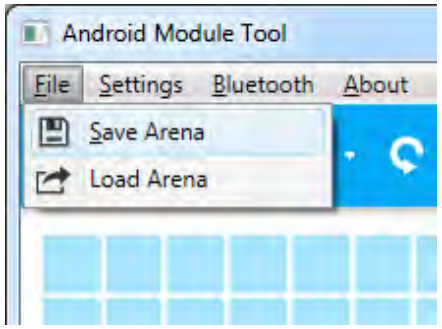


Adjust robot rotation

- Rotate left to next 90° interval
- Rotate right to next 90° interval
- Manually adjust robot orientation starting from North, clockwise (i.e. at 0°, robot points towards top of the virtual arena. Increasing the degree value will make the robot rotate clockwise.)

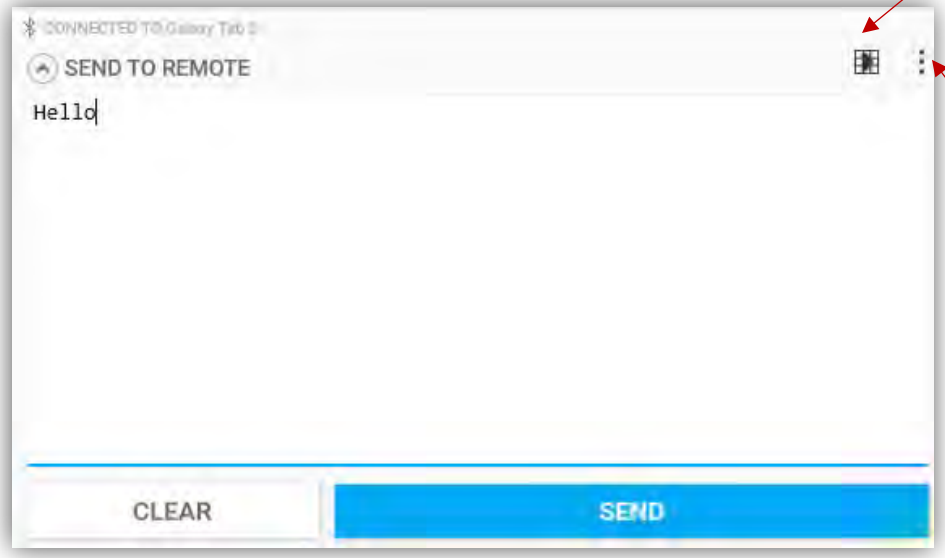
Virtual robot (Facing North)

Save arena layout with robot as .txt file, allowing you to load it later.



Send Text to Remote Device

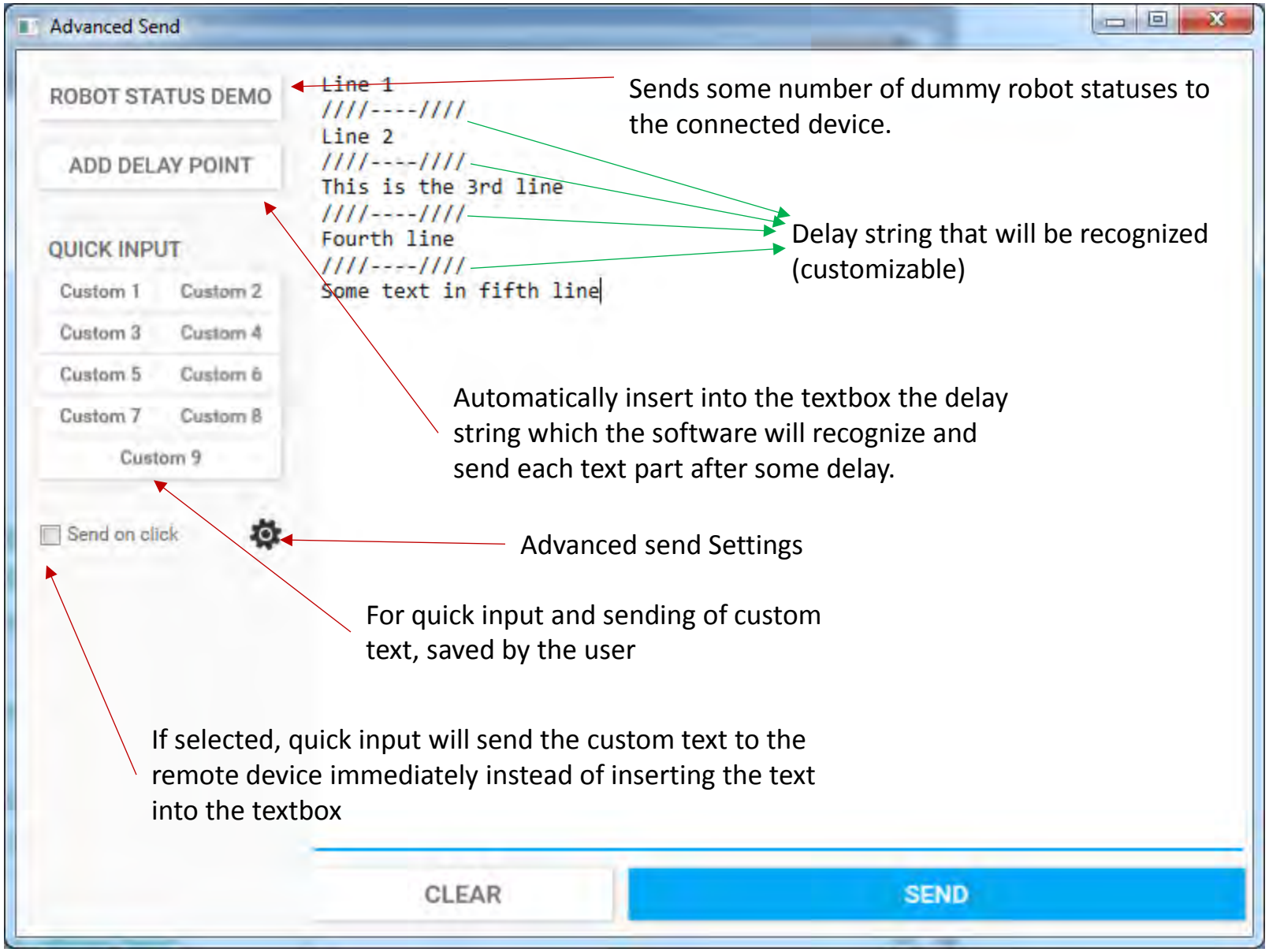
- If toggled, any changes to the virtual arena, including the robot, will let the software to send such arena information to the remote device.
- This will allow you to debug the display of the arena and the robot on your app.
- Format in which the remote device receives the arena information can be edited with custom scripts (Discussed later)



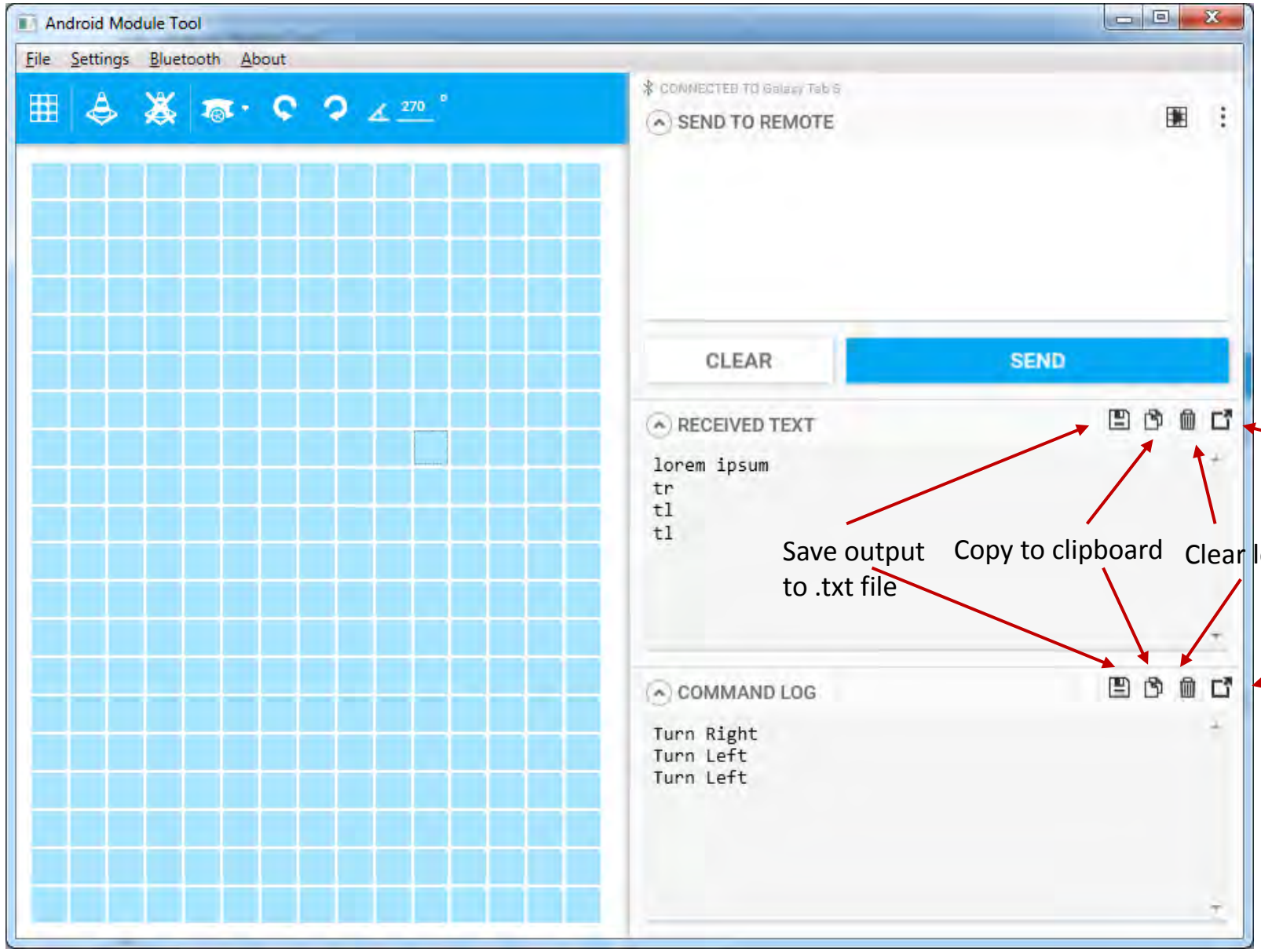
Click to show advanced send functions AND
have more space to concentrate on sending text, especially when testing with long string of text.

Connection to device must be established first to use this.

Send Text to Remote Device (Advanced Send Window)



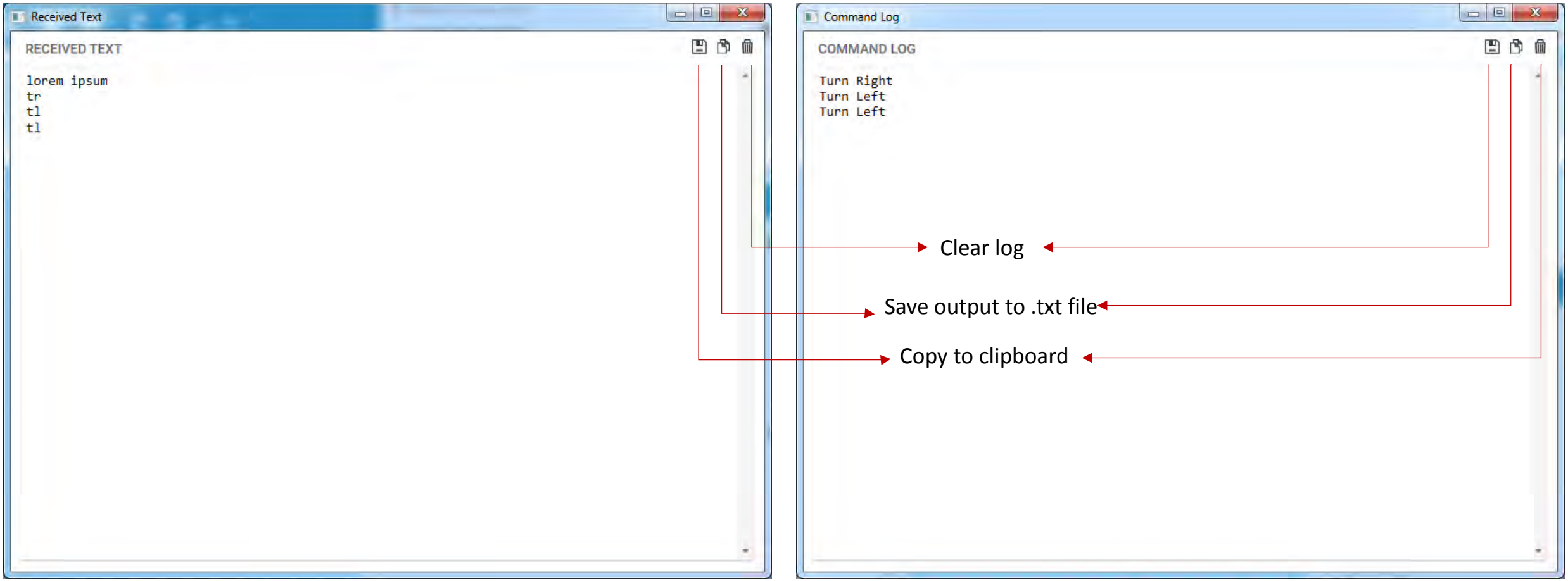
Received Text & Command Log



Received Text
Log of text that the remote device had sent.

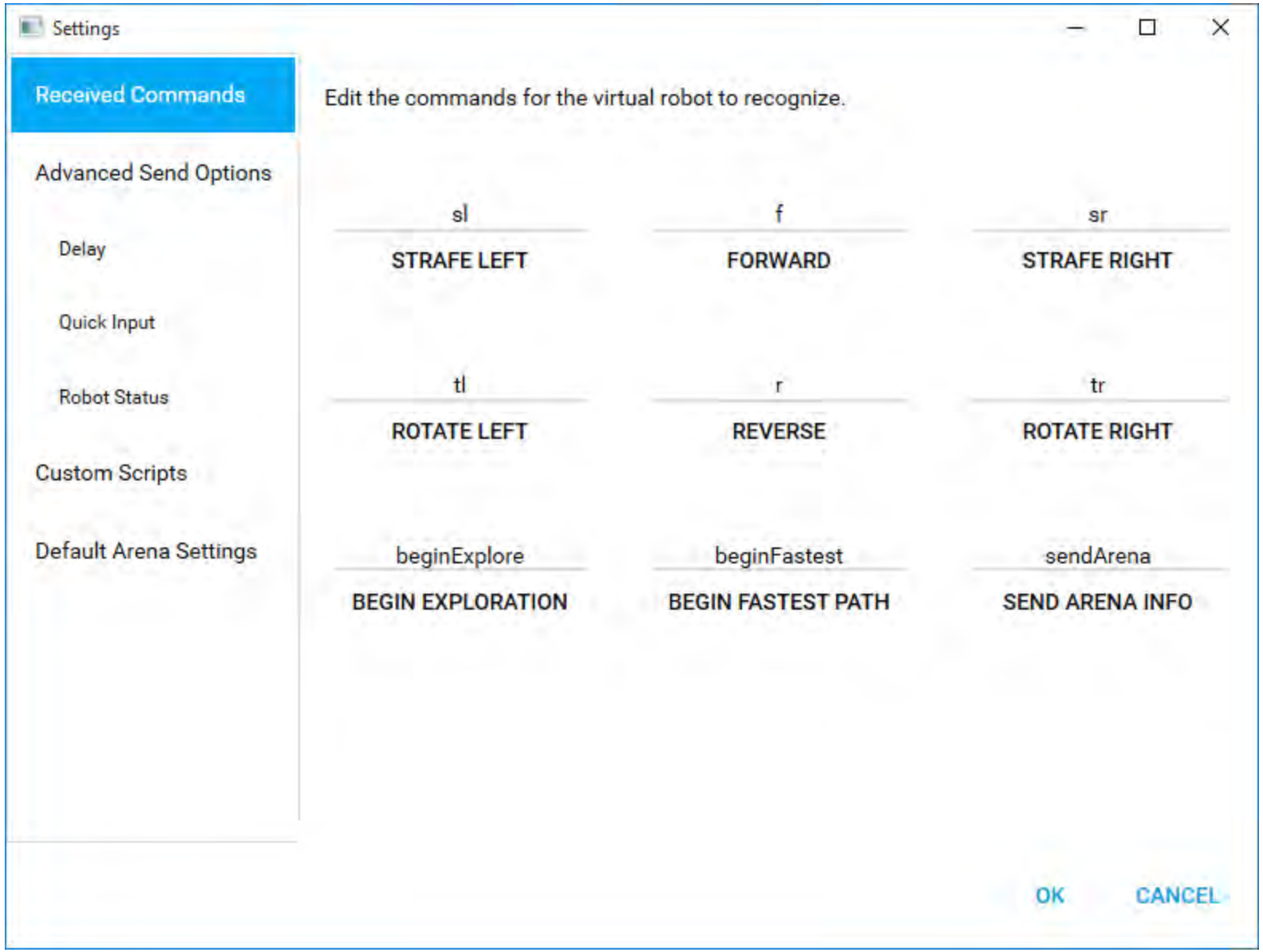
Command Log
If any received text is a command, the corresponding command will be displayed. The virtual robot will also react accordingly to the command.

Received Text & Command Log (Expanded Window)



Expanded window to concentrate on receiving text/ checking commands, especially when testing with long string of text.

Settings

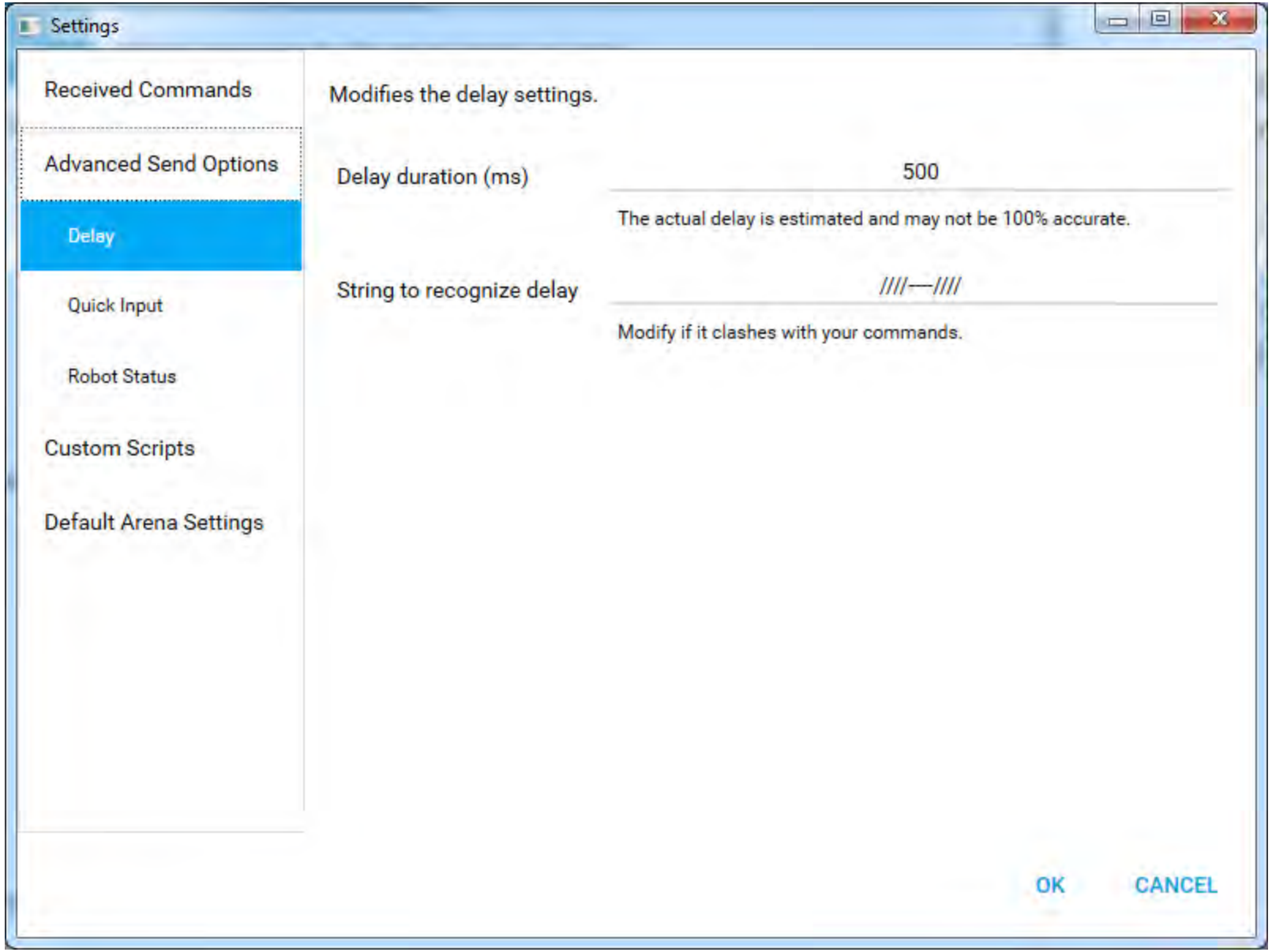


Allow you to modify the commands for the virtual robot to recognize.

For example if the command for forward is set to "*move:forward*", whenever the software receives this text, the virtual robot will move forward in the arena.

The "Send Arena Info" command will let the software send the arena obstacle and robot position information to the android remote.

Settings

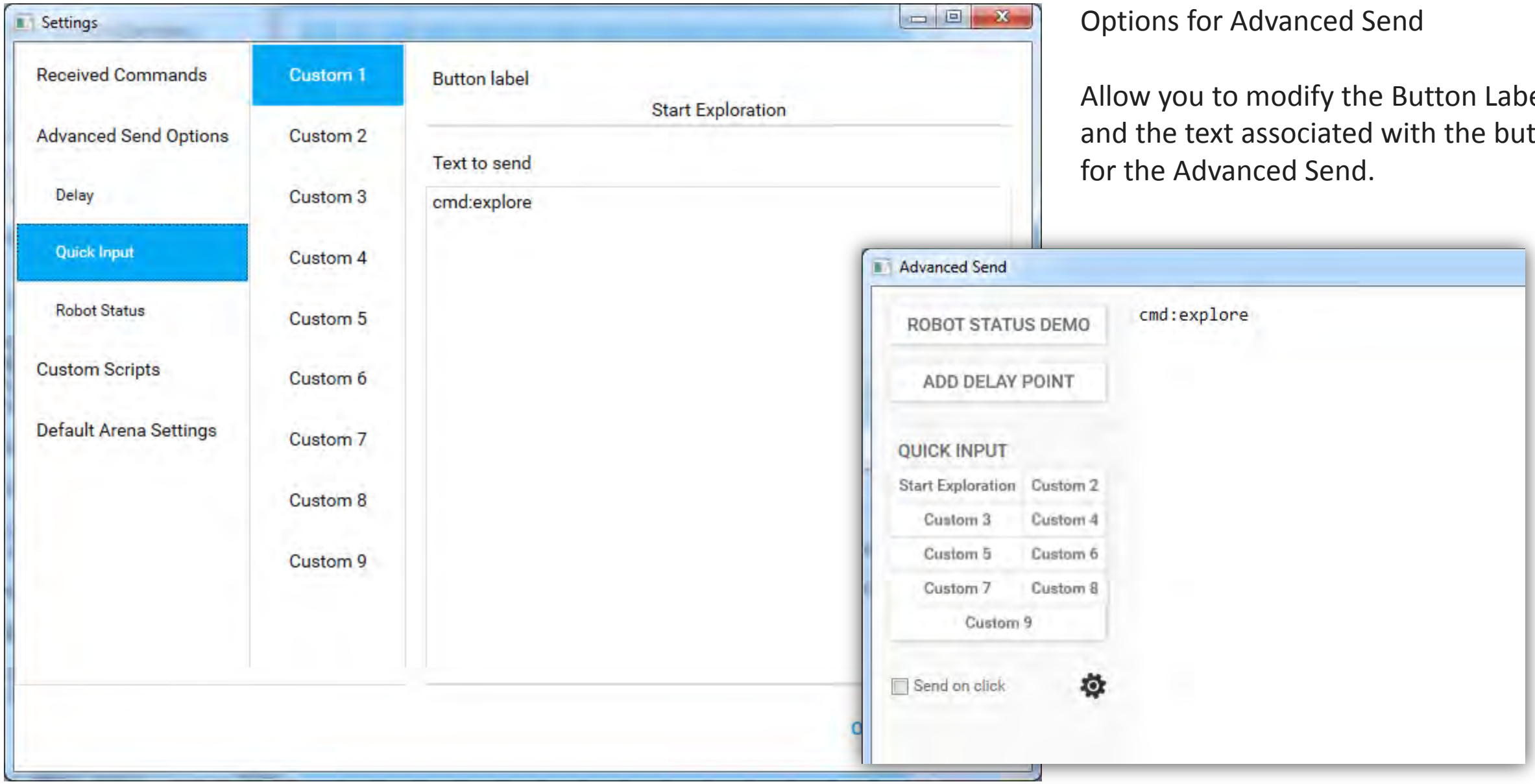


Options for Advanced Send

Allow you to modify the delay duration for the advanced send.

Allow you to modify the string to recognize as a delay, especially if it clashes with the team's commands.

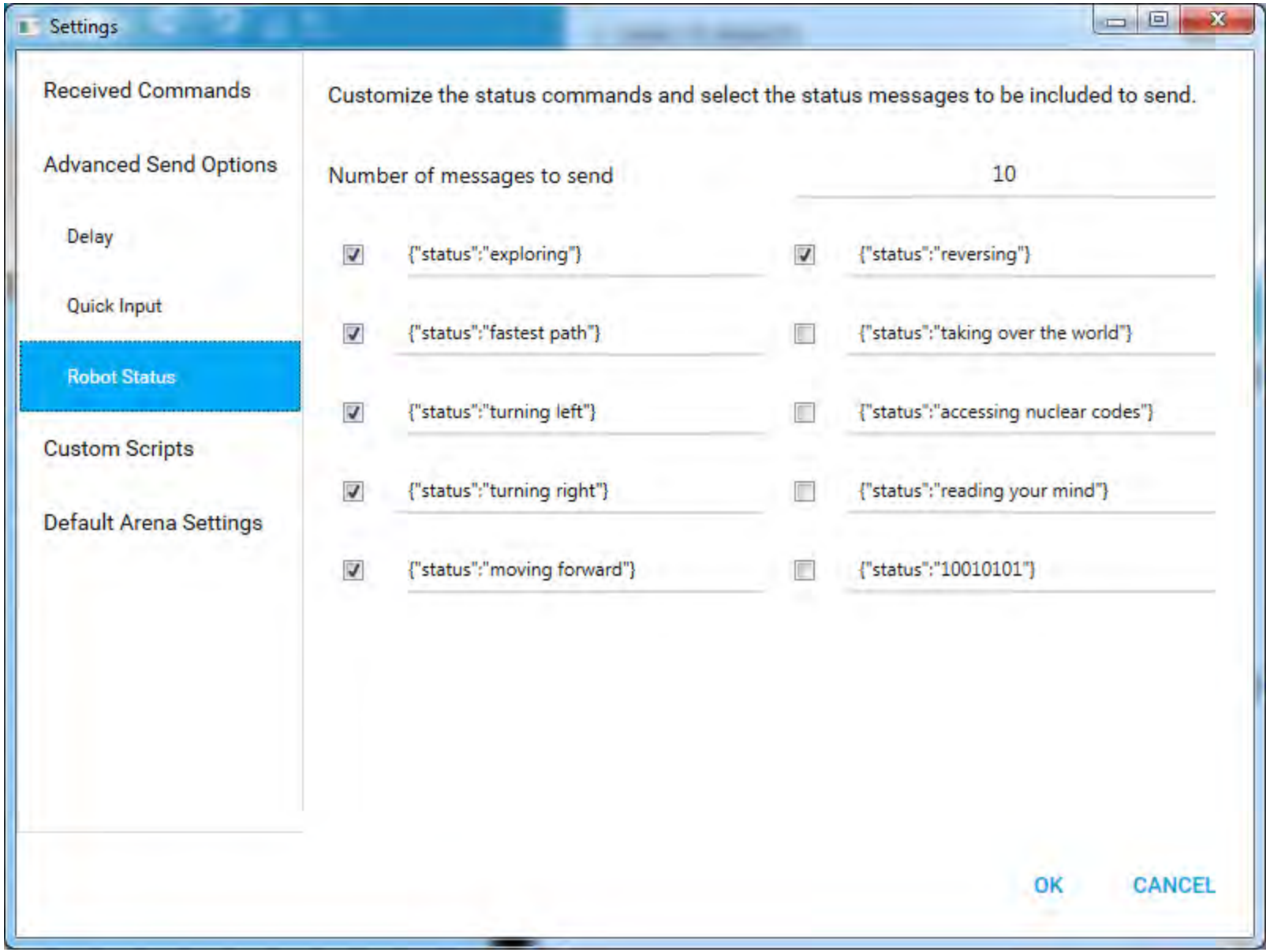
Settings



Options for Advanced Send

Allow you to modify the Button Labels and the text associated with the button for the Advanced Send.

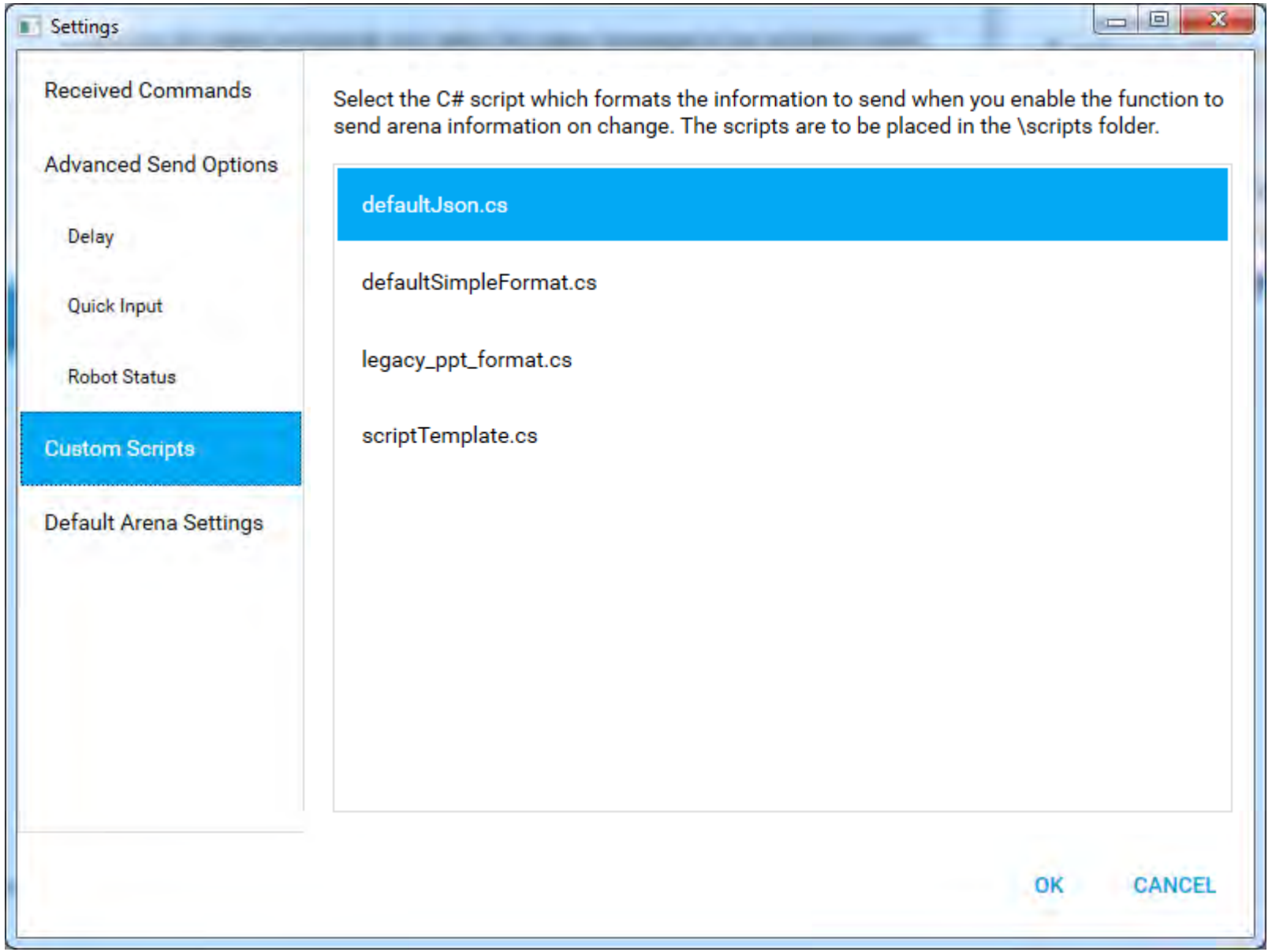
Settings



Options for Advanced Send

Customization of the robot status demo. This will remove the format restrictions as the software will send only robot status message specified by the user.

Settings



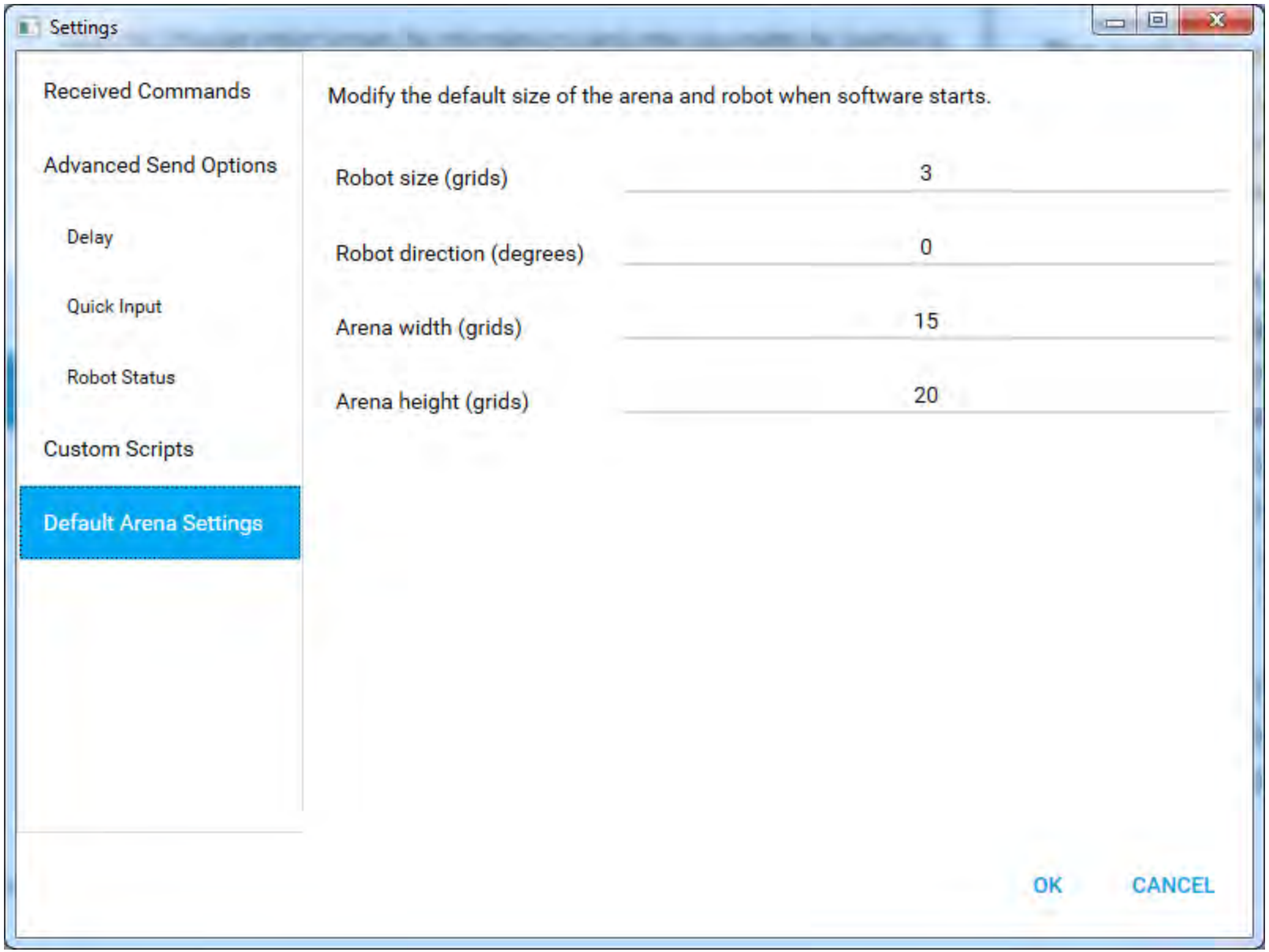
Custom scripts to format arena information that would be sent to remote device.

Instructions on how to create your own scripts are detailed in the 3 example files in “scripts” folder.

“scriptTemplate.cs” is a template file with comments detailing on how to use and implement it.

“defaultJson.cs” and “defaultSimpleFormat.cs” are example implementations that you can take reference from or to use.

Settings

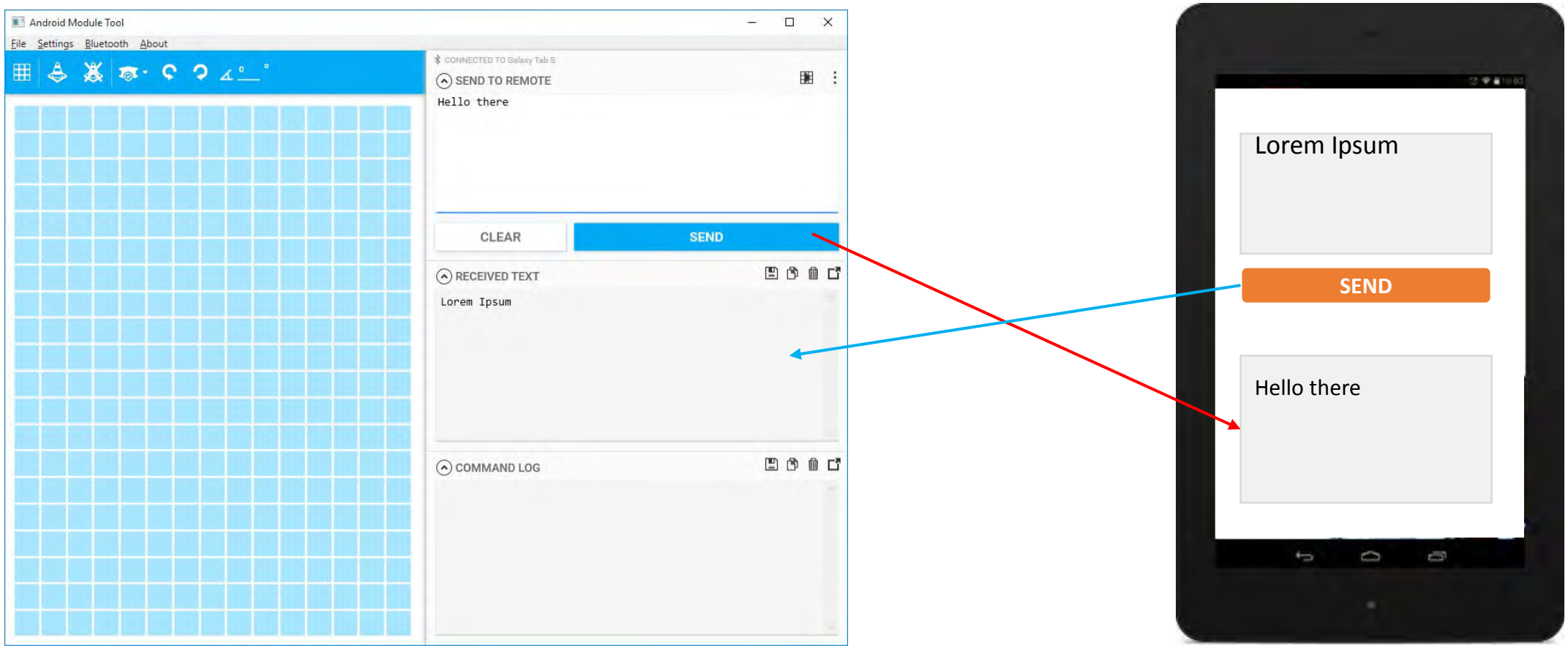


Default Arena Settings

Adjust the default arena settings that will load on start.

Clearing checklist - C1

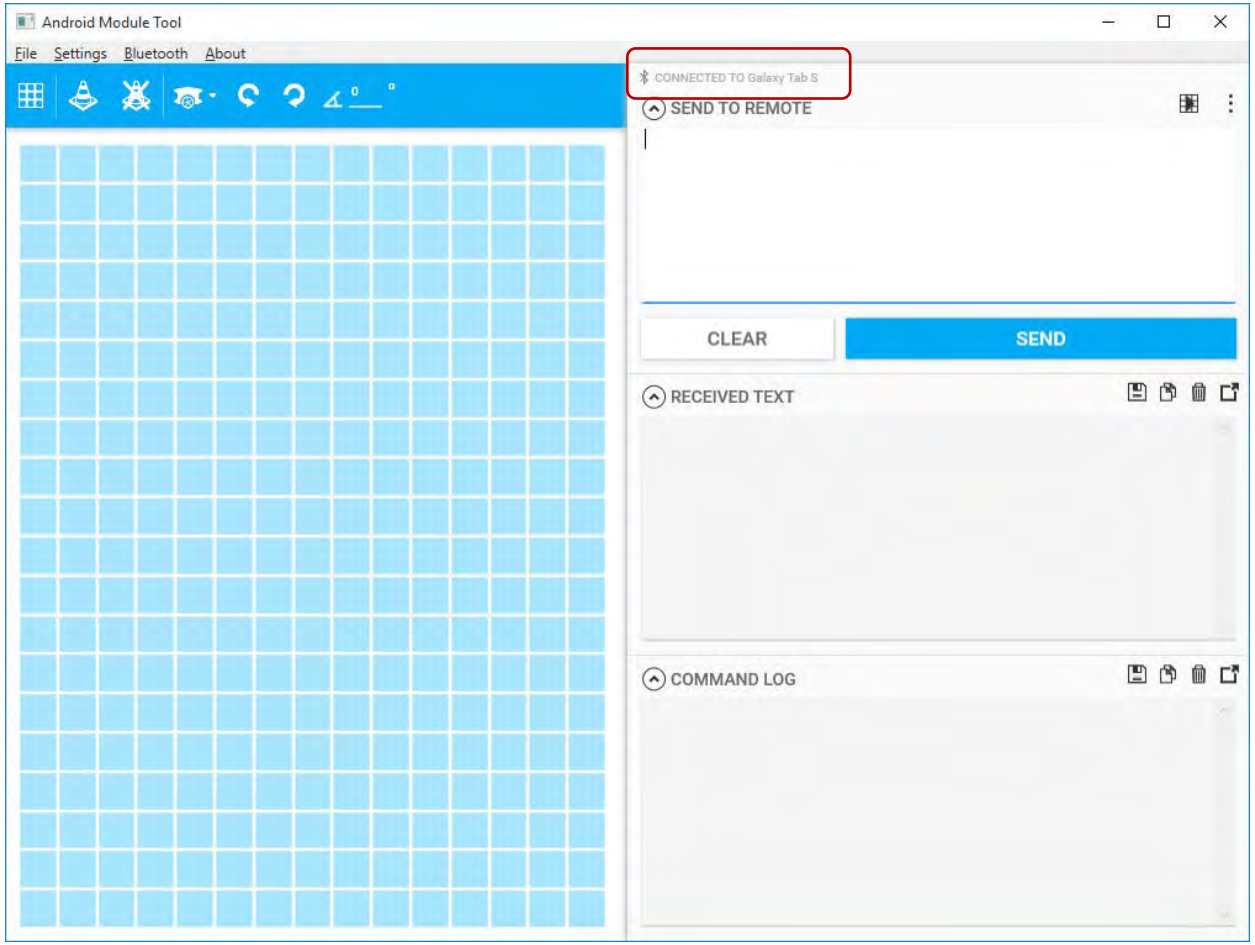
- The Android application (AA) is able to transmit and receive text strings over the Bluetooth serial communication link.



Clearing checklist – C2

- Functional graphical user interface (GUI) that is able to initiate the scanning, selection and connection with a Bluetooth device.

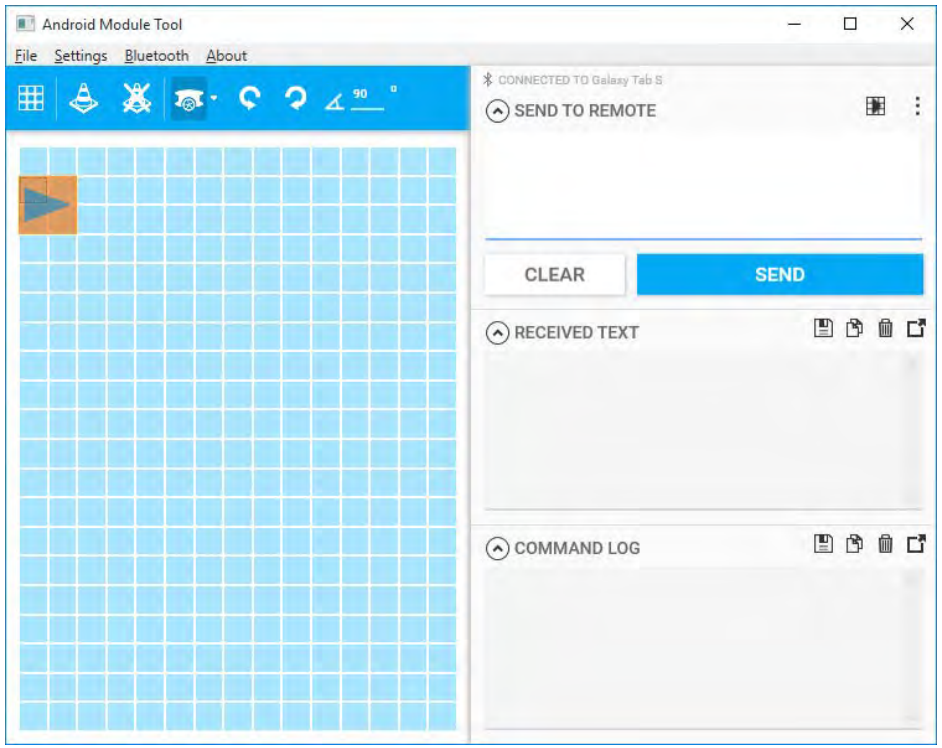
The software can act as Bluetooth server



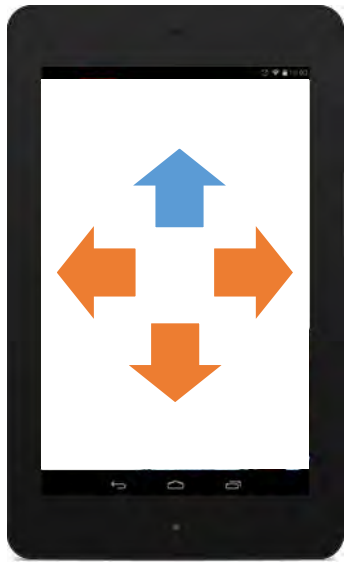
Clearing checklist – C3

- Functional GUI that provides interactive control of the robot movement via the Bluetooth link

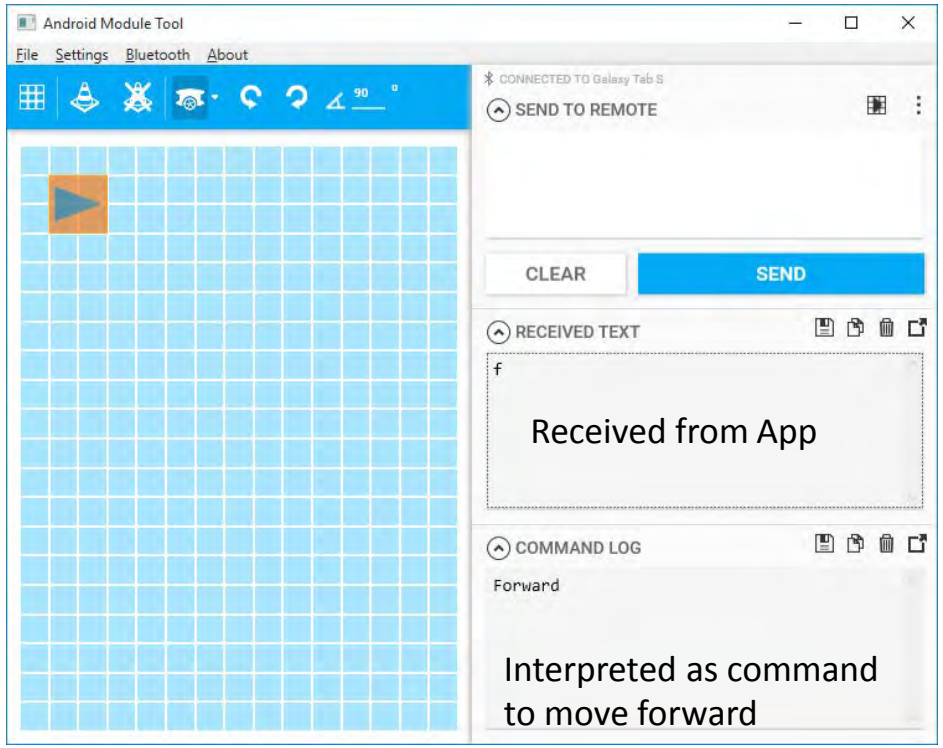
Initial state



Move forward

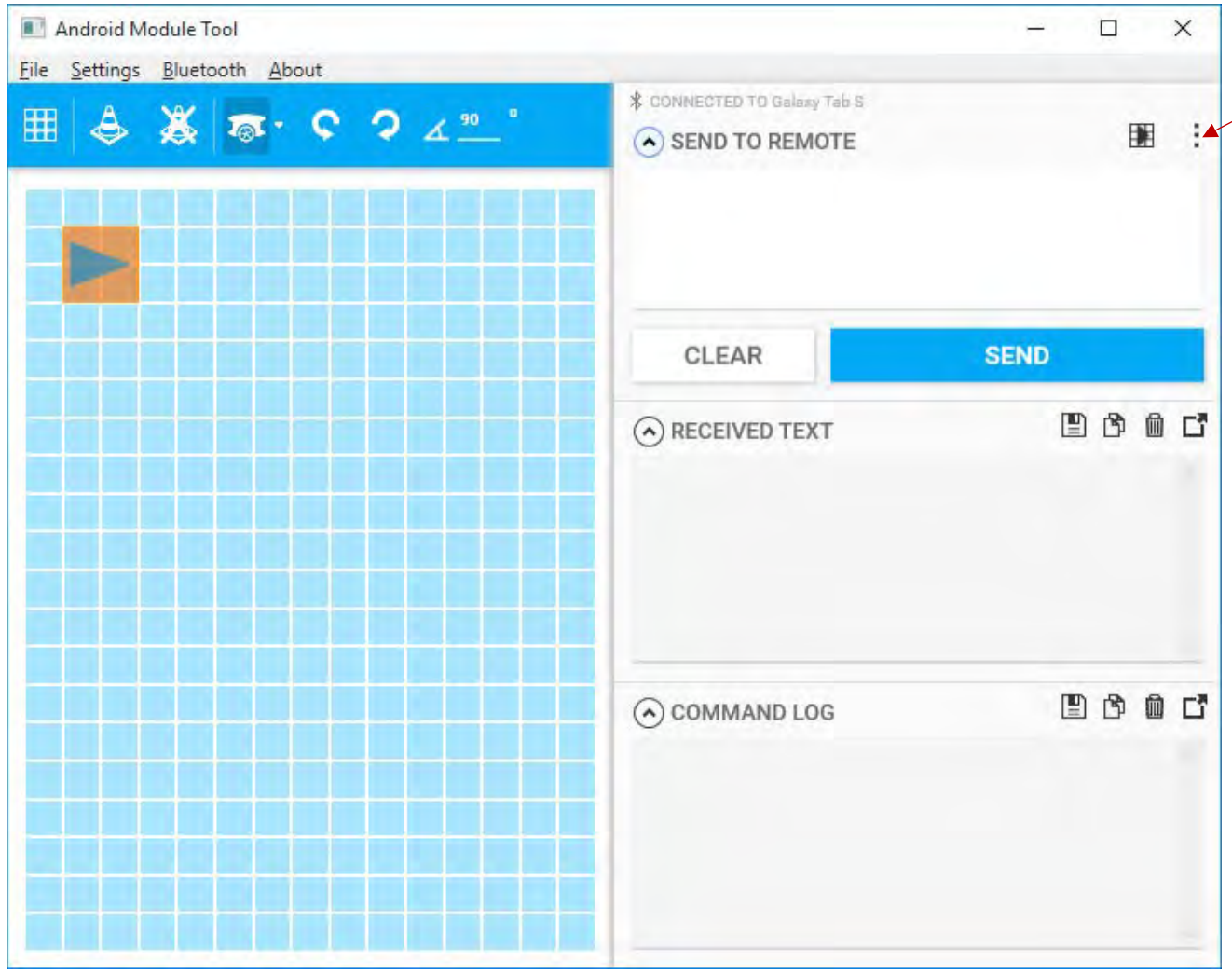


Robot position shifted



Clearing checklist – C4

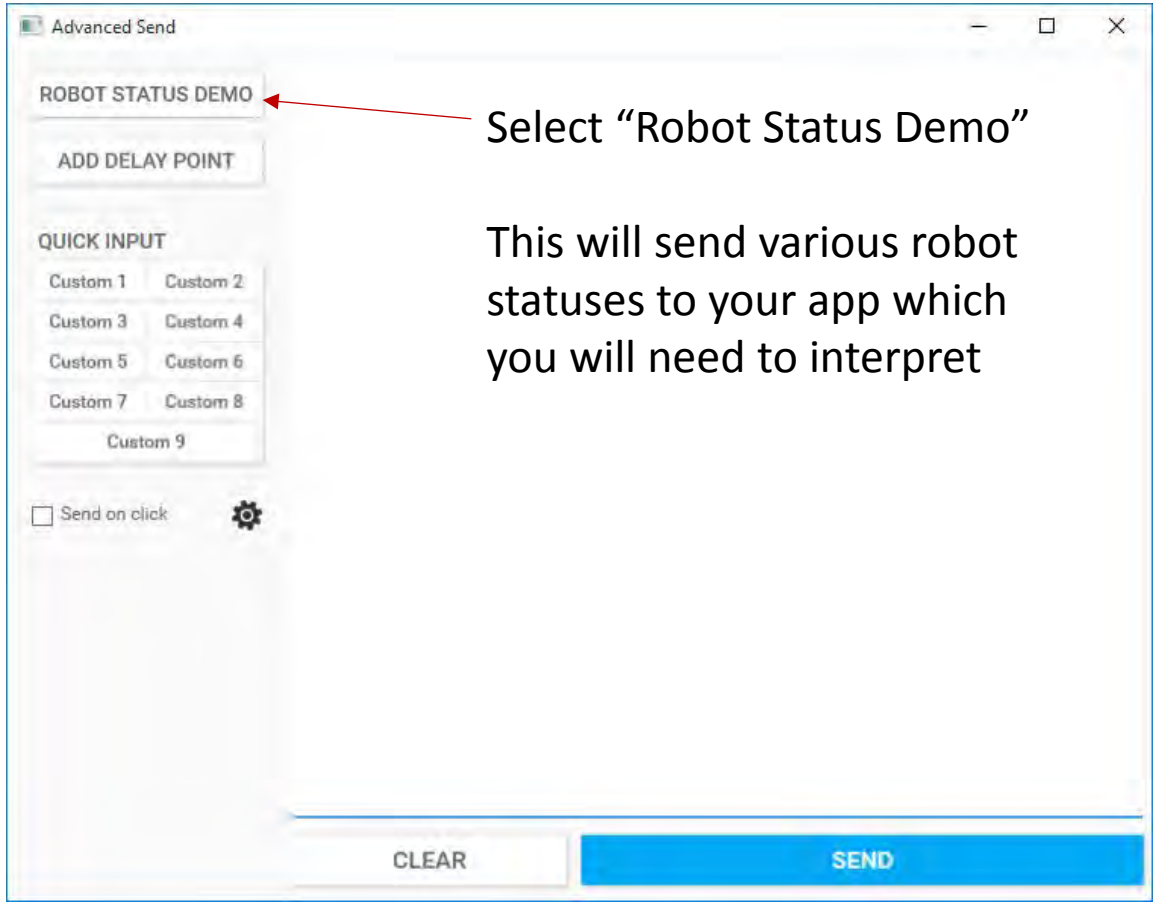
- Functional GUI that indicates the current status of the robot



Select this to use advanced send functions

Clearing checklist – C4

- Functional GUI that indicates the current status of the robot

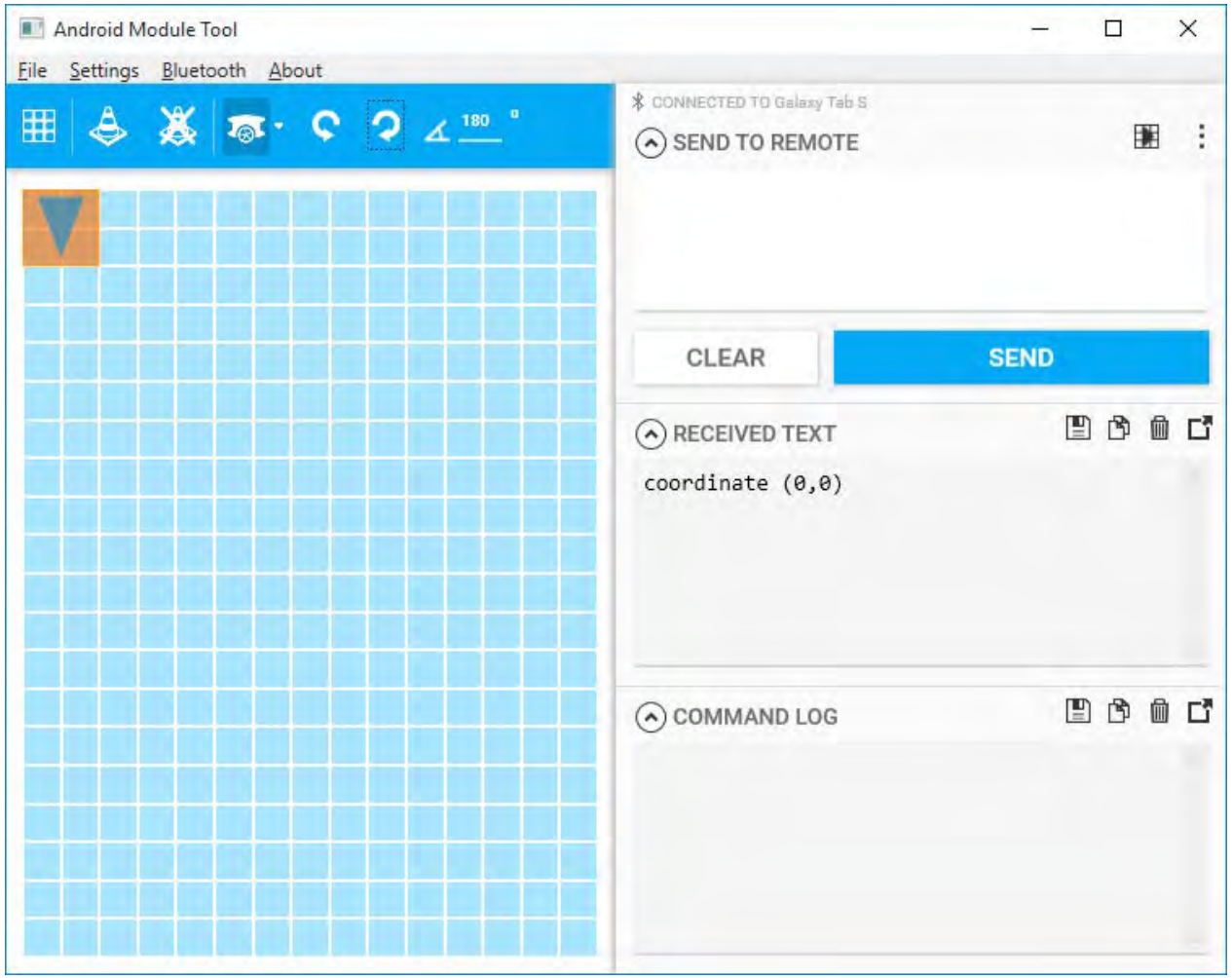


Select “Robot Status Demo”

This will send various robot statuses to your app which you will need to interpret

Clearing checklist – C5

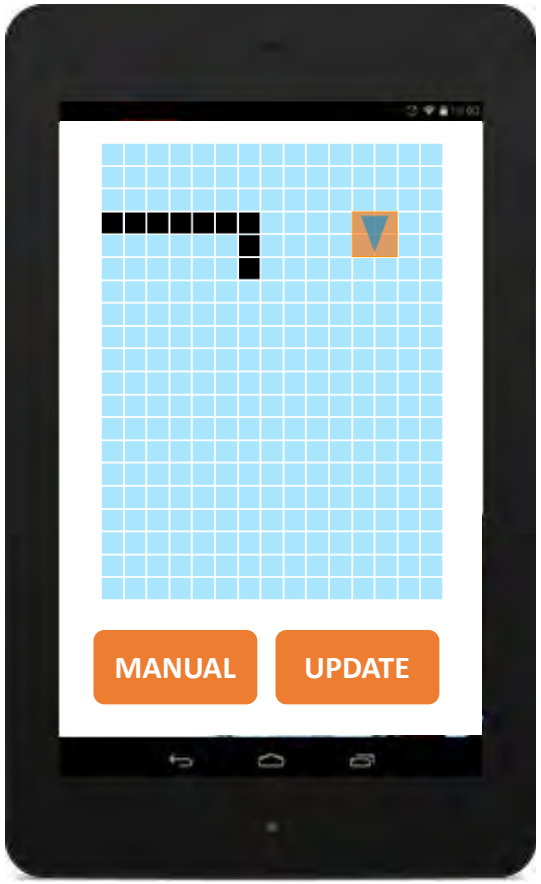
- Update Robot start coordinates

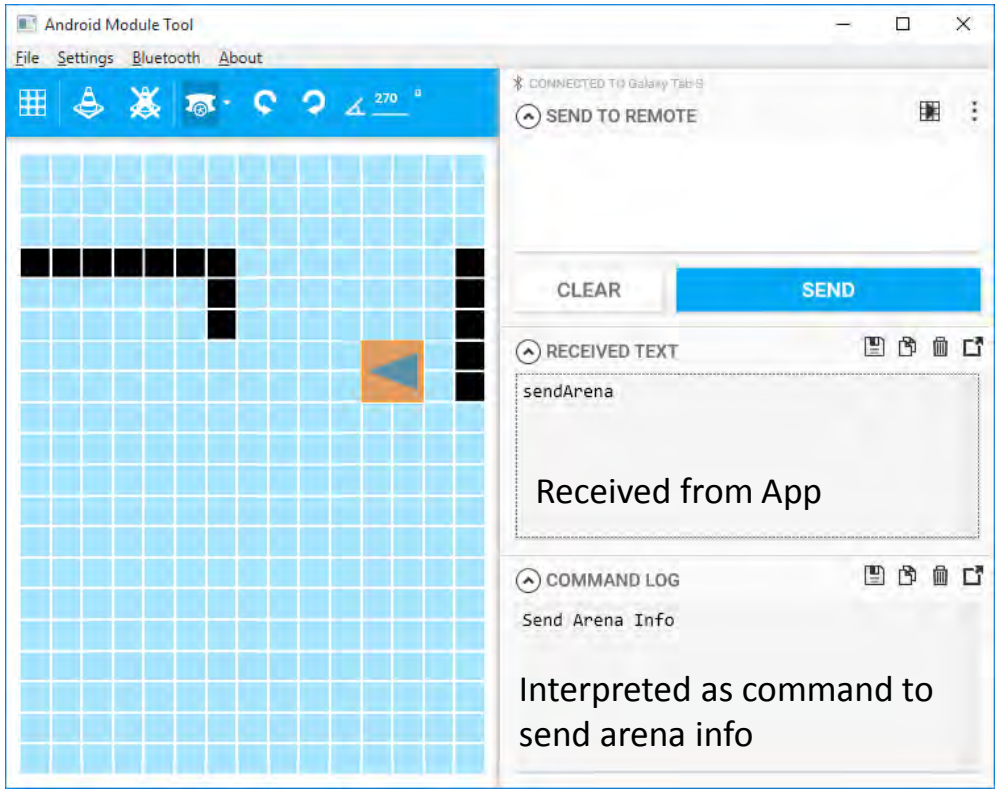


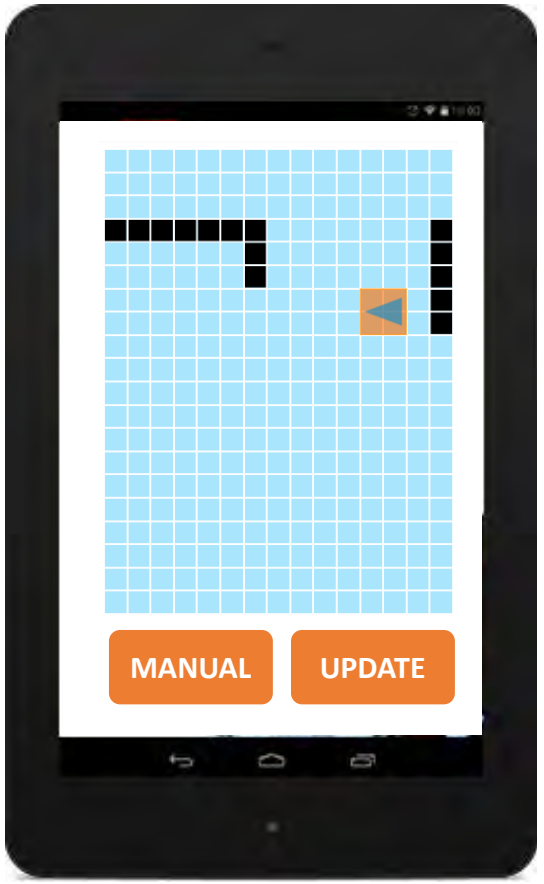
Clearing checklist – C7

- Functional GUI that provides the selection of Manual or Auto updating of graphical display of the maze environment.

1. Add obstacles and/or change position of robot on tool
2. Send command to request for arena info on app (Manual or Auto)



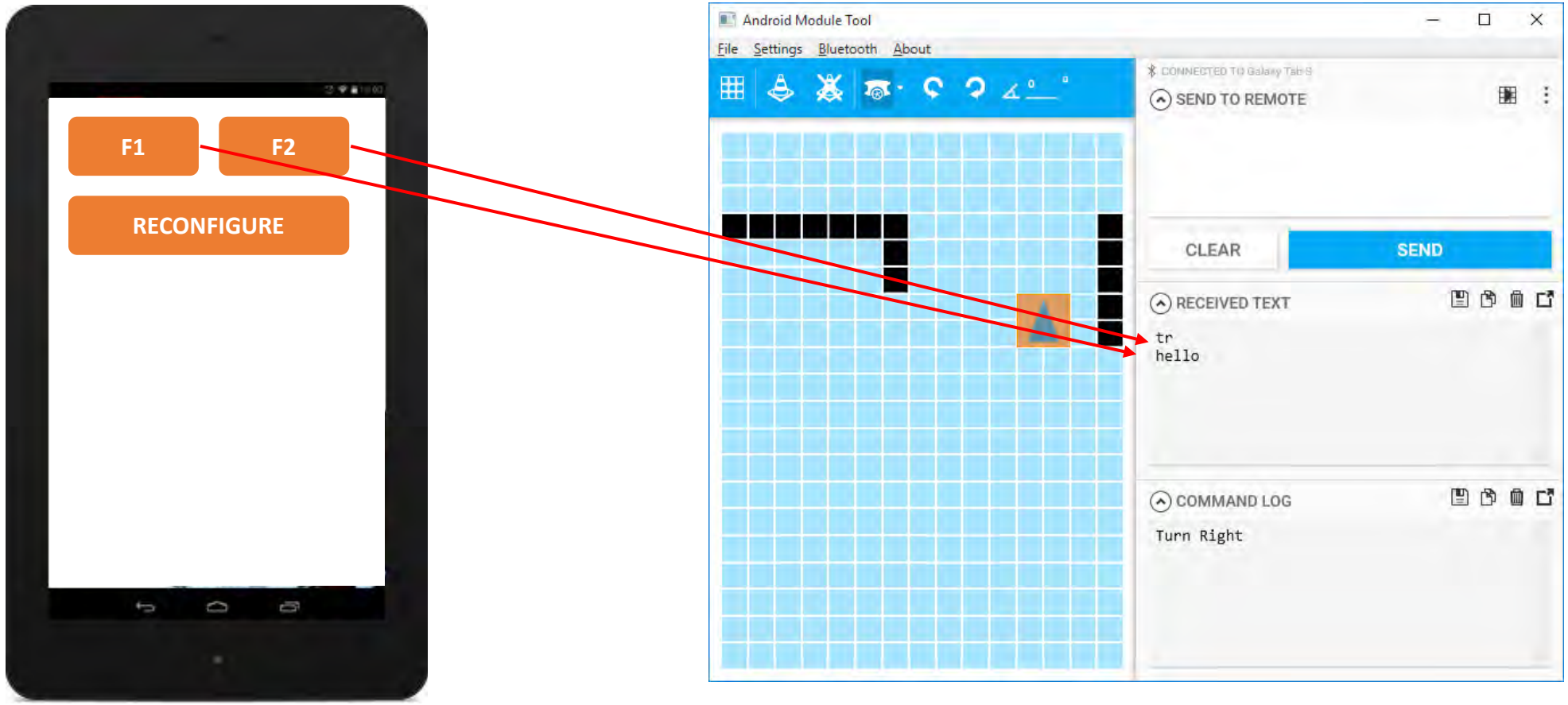




Clearing checklist – C8

- Functional GUI that provides two buttons that supports persistent user reconfigurable string commands to the robot.

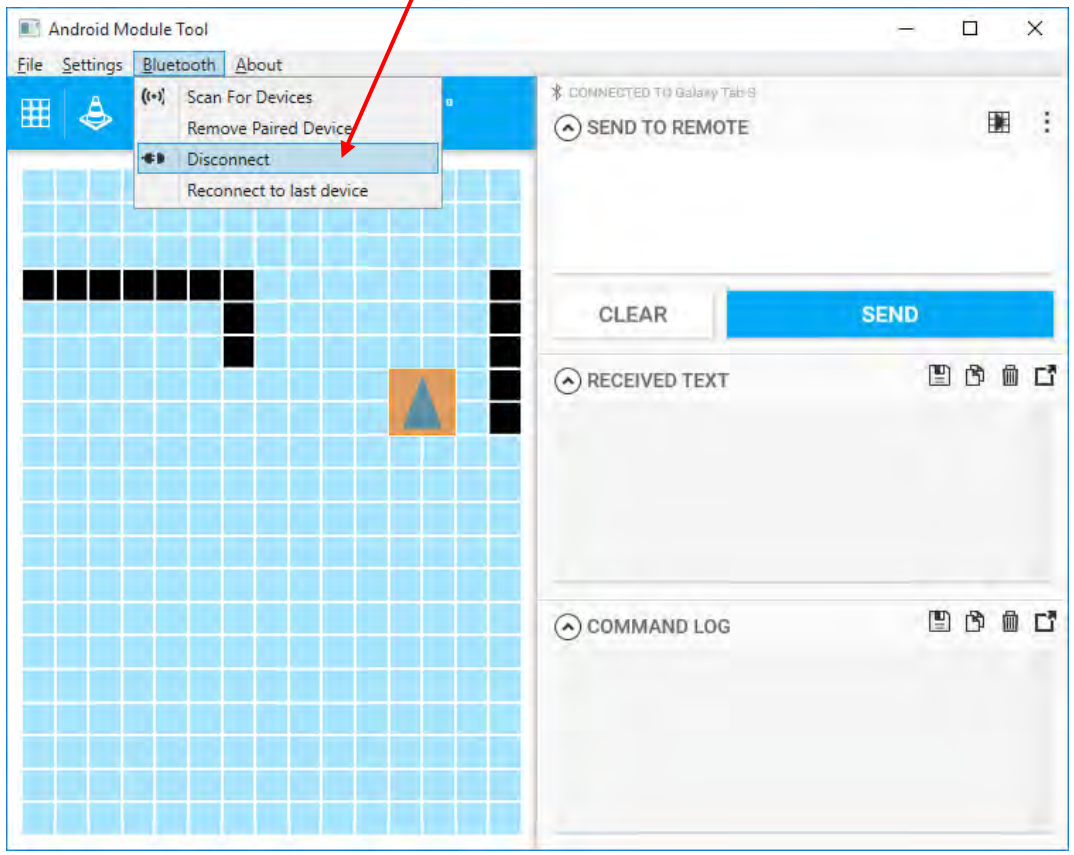
When button is pressed



Clearing checklist – C9

- Robust connectivity with Bluetooth device.

Disconnect



Select “Reconnect to last device”
Ensure that there is connection to your app after this.