

MDP

Android Remote Controller

Module Briefing

by
A/P Goh Wooi Boon

29 Jan 2016

Contents

- Introduction
- Project Deliverable Checklist
- Extending beyond the minimal deliverables
- How to get started?

Introduction

- The objective of the Android Remote Controller Module in the MDP is to introduce practical issues related to:
 - **Mobile Computing**
 - **Human Computer Interaction**



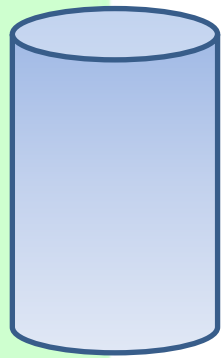
Nexus 7 Tablet

What will you be doing?

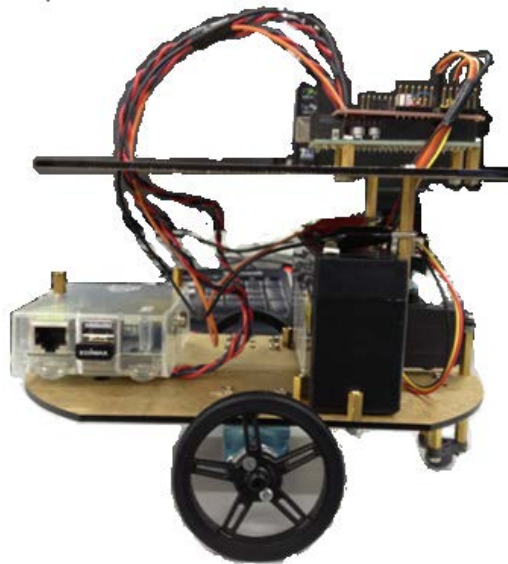
- Develop mobile apps on an Android-powered device.
- Design and develop graphical user interface-based apps.
- Implement wireless connectivity between Bluetooth-enabled devices.
- Design and implement graphical displays in your Android app.

How do you contribute to your team?

- Your Android tablet will be the **wireless remote controller** device for your team's robotic system.
- It will issue commands to robot to begin various manoeuvres in maze during the competition.
- It will allow the team to visualize the current status of maze and robot.

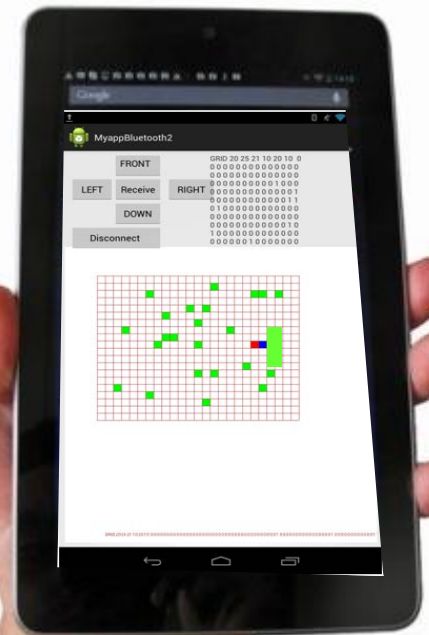


Obstacles
in maze



Robotic Subsystem

Bluetooth
link



Nexus 7 Tablet

How is this module assessed?

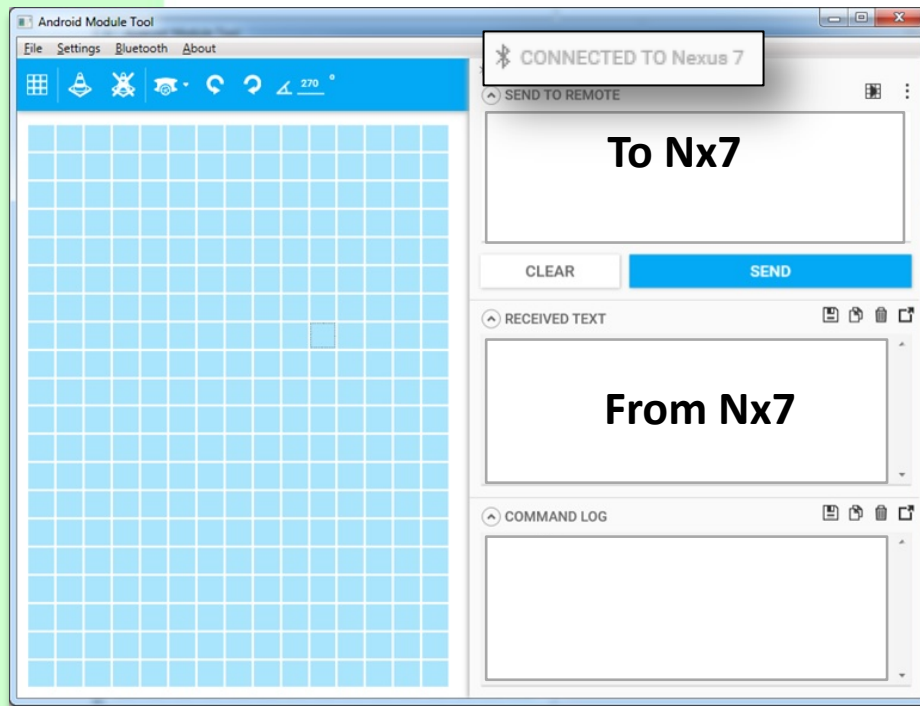
- The assessment of the Android remote controller module (ARCM) can be done independently of other modules.
- The **Project Deliverable Checklist** (20%) has a section on the Android remote controller module (section C).
- The deliverable checklist (section C) represents the **minimum** implementation you should undertake for this module.
- However, the ACRM team must **work closely** with the rest of the teams doing the other modules to ensure a **smooth integration** at a later stage.
- This is necessary for the team to participate in the **leaderboard competition**.

ARCM Deliverable Checklist

C.1 Your Android app is able to **transmit** and **receive** text string over the Bluetooth serial link.

AMDtool.exe

C.2 Your Android app GUI can initiate **scanning**, **selection** and **connection** with Bluetooth device.



AMD Tool



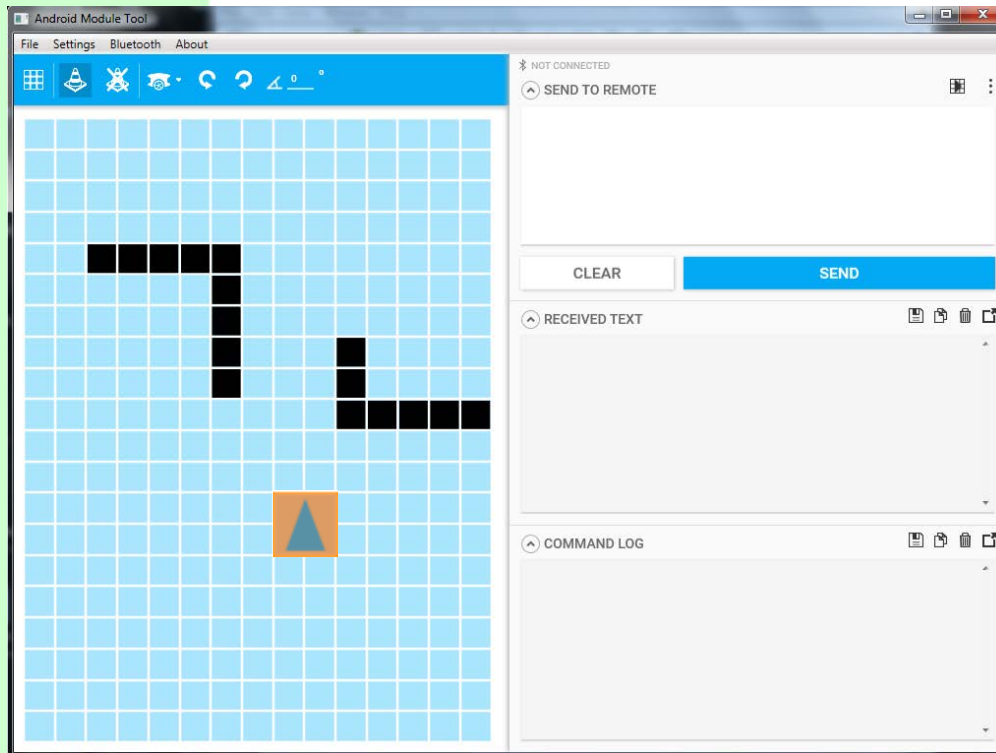
Bluetooth
link



Nexus 7 Tablet

ARCM Deliverable Checklist

- C.3** Your Android app GUI provides interactive control of **robot movement** (via Bluetooth link).
- C.4** Your Android app GUI provides indication of the **current status** of the robot.



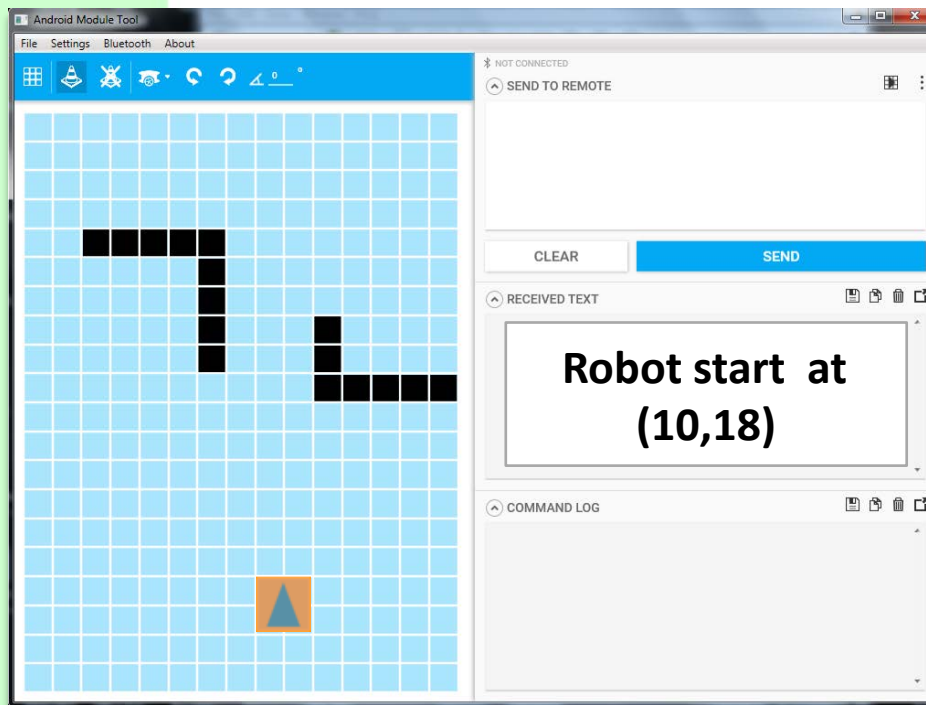
Bluetooth
link



Nexus 7 Tablet

ARCM Deliverable Checklist

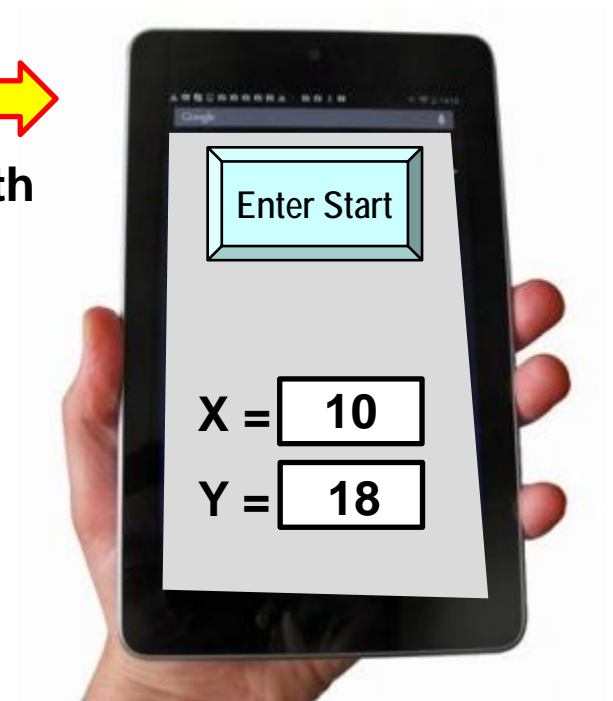
C.5 Update robot start coordinates.



AMD Tool



Bluetooth
link



Nexus 7 Tablet

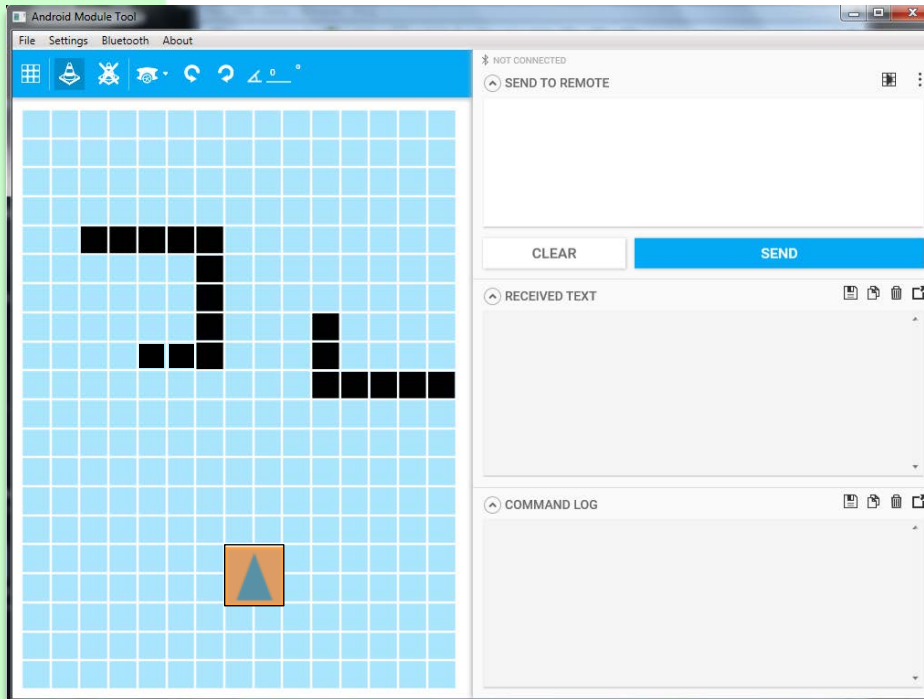
ARCM Deliverable Checklist

C.6 Your Android app is able to **display** the maze with its current known obstacles and robot.

C.7 Android app GUI provides selection of **Manual /Auto** display update mode.

Important

The map display on the Android tablet is required in the competition.



AMD Tool



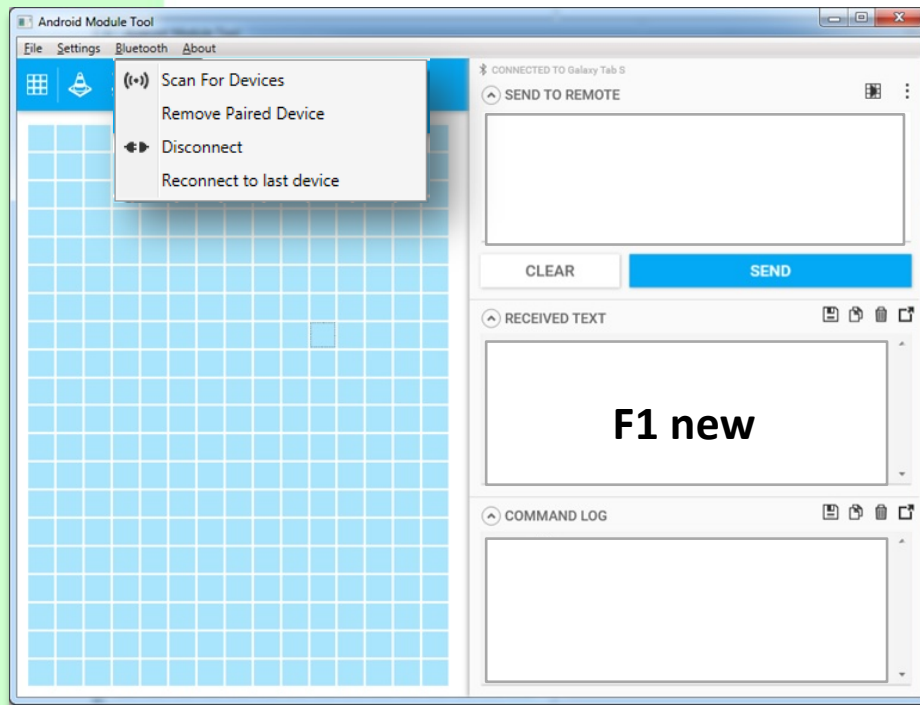
Bluetooth
link



Nexus 7 Tablet

ARCM Deliverable Checklist

- C.8** Your Android app GUI provides two buttons that support **persistent user reconfigurable** string commands to robot.
- C.9** Your Android app provides **robust Bluetooth** connectivity.



AMD Tool



Bluetooth
link



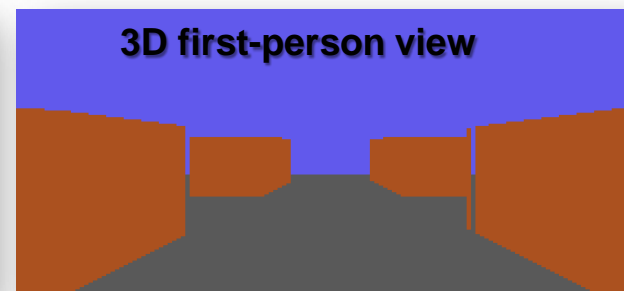
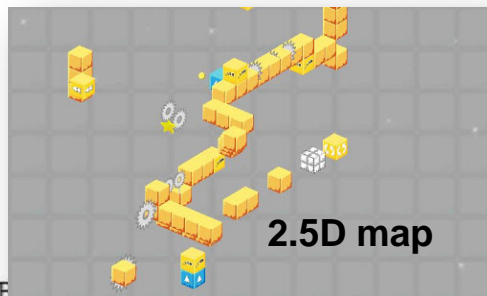
Nexus 7 Tablet

Extending beyond the minimal deliverables

- Teams are encouraged to go beyond the requirements stated by the project deliverable checklist.
- Marks will be given for special achievements in:
 - Video report submission (under the Content category)
 - Group project wiki

Some ideas to explore (final Android checklist item):

- **Novel robot movement control** – Using touch gesture, tilt sensing (accelerometer) or clever combination of these sensors.
- **Interesting graphical display** – Use 3D view from robot or 2.5D display of grid map, obstacles and robot.



How to get started?



Get started
with Android
App
Development
.pptx



Tutorials on
Android
Apps.pptx

- Install the Android Development Tools (ADT).
- Review ARCM checklist and discuss how you want to **split the responsibility** for the various functionalities.
- **Learn by experimenting** with relevant online tutorials and sample codes.
- Before you start coding, do paper design of the **number of activities** you need and the **GUI layout design** for each.
- To allow easier code integration later, **agree and assign** appropriate names to variables, widget ids (e.g. for buttons, text box, etc) and associated routine names and functionality.
- The person doing the Bluetooth module will probably need to use the Android **tablet** and the other doing graphics can use the Android **emulator** during early coding phase.

Acknowledgements

- The Android Module Debugger (AMD) Tool was developed by Lee Shei Pin (CSC Year 4 - 2015) as part of his FYP.
- The current version has gone through many iterations of improvements from previous students who have used it and provided feedback.
- I like to acknowledge Shei Pin's contributions in developing this software tool to make your MDP checklist evaluation much easier for both you and the MDP supervisors.

The End

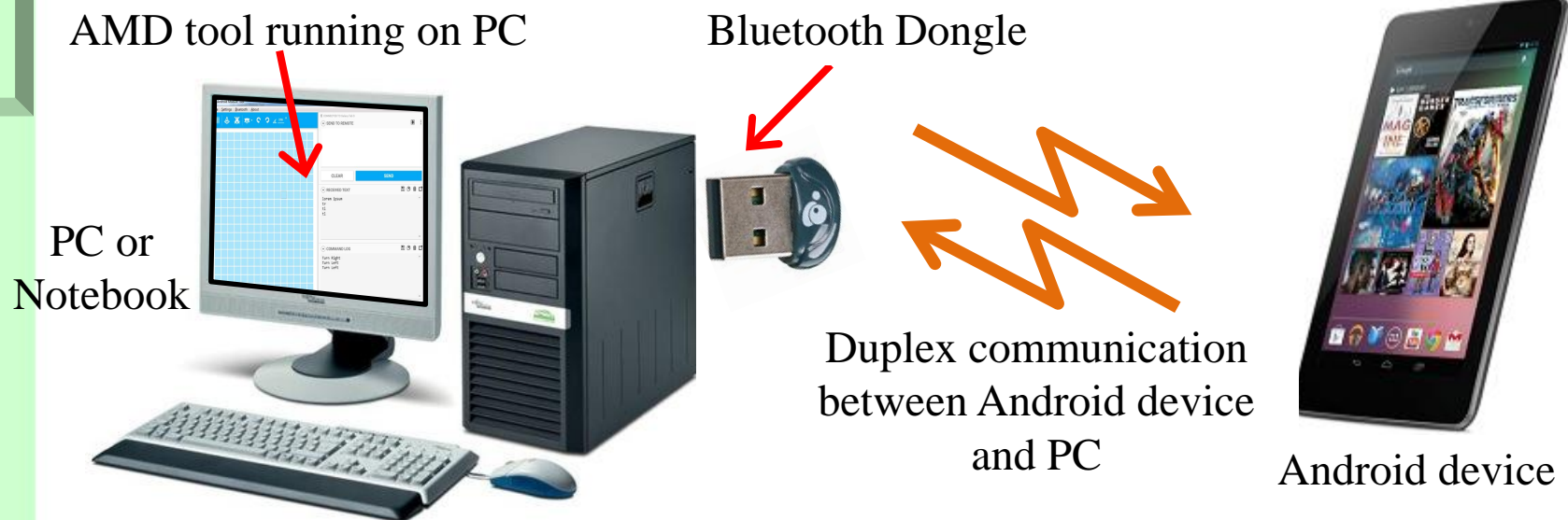
Have an Android hAPPy experience!

A/P Goh Wooi Boon

Using the AMD Tool

- The Android Module Debugger (AMD) tool (**AMDtool.exe**) can be downloaded into your PC or Notebook from the Edventure site: **Technical Materials/Android**. (Note: Unzip the downloaded file into a new folder)
- This Windows application program can be used to debug your Bluetooth serial communication during development and to verify checklist functionalities related to the Bluetooth link.

Run
MDP-AMD
Program



Note: For AMDtool clarifications and bug reporting, please contact Lee Shei Pin at LEES0147@e.ntu.edu.sg