Mobile Applications for Sensing and Control

Sep Makhsous

Week 5

Bluetooth

Android Bluetooth APIs

Key Components:

- BluetoothAdapter: The phone's Bluetooth radio
- BluetoothDevice: Remote device
- BluetoothSocket: Communication channel
- BluetoothManager: Manages multiple devices (Android 4.3+)

First Step - Permission

- Similar to camera we need to add permission in the Manifest file

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
```

- From Android 12+, BLUETOOTH_CONNECT and BLUETOOTH_SCAN are mandatory.
- Also need Location permissions for scanning!

Basic Bluetooth Flow

- Check if Bluetooth is supported
- Request user to enable Bluetooth
- Scan for devices
- Show device list
- Connect to a selected device
- Send and receive data

Setting up BluetoothAdapter

```
// Try to get the device's Bluetooth adapter (the phone's Bluetooth radio)
val bluetoothAdapter: BluetoothAdapter? = BluetoothAdapter.getDefaultAdapter()

// Check if the device supports Bluetooth
if (bluetoothAdapter == null) {
    // If adapter is null, this device doesn't support Bluetooth
    // Show a message to the user or disable Bluetooth-related features
}

// Check if Bluetooth is enabled
if (!bluetoothAdapter.isEnabled) {
    // Bluetooth is supported but not enabled; prompt the user to turn it on
    val enableBtIntent = Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE)
    startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT)
}
```

Discovering Devices

```
// Create a BroadcastReceiver to handle when a new Bluetooth device is found
val receiver = object : BroadcastReceiver() {
    override fun onReceive(context: Context, intent: Intent) {
        // Check if the received broadcast event is that a Bluetooth device was found
        if (BluetoothDevice.ACTION_FOUND == intent.action) {
            // Extract the BluetoothDevice object from the Intent
            val device: BluetoothDevice? = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE)

            device?.let {
                // If the device is not null, add it to your device list (e.g., to show in a RecyclerView)
                // You would typically update your UI here or notify your adapter
 }}}}
```

Listing Devices in UI

- Use:
  - RecyclerView to list found devices
  - Show name + MAC address

data class BluetoothDeviceItem(val name: String, val address: String)

Connecting to a Device

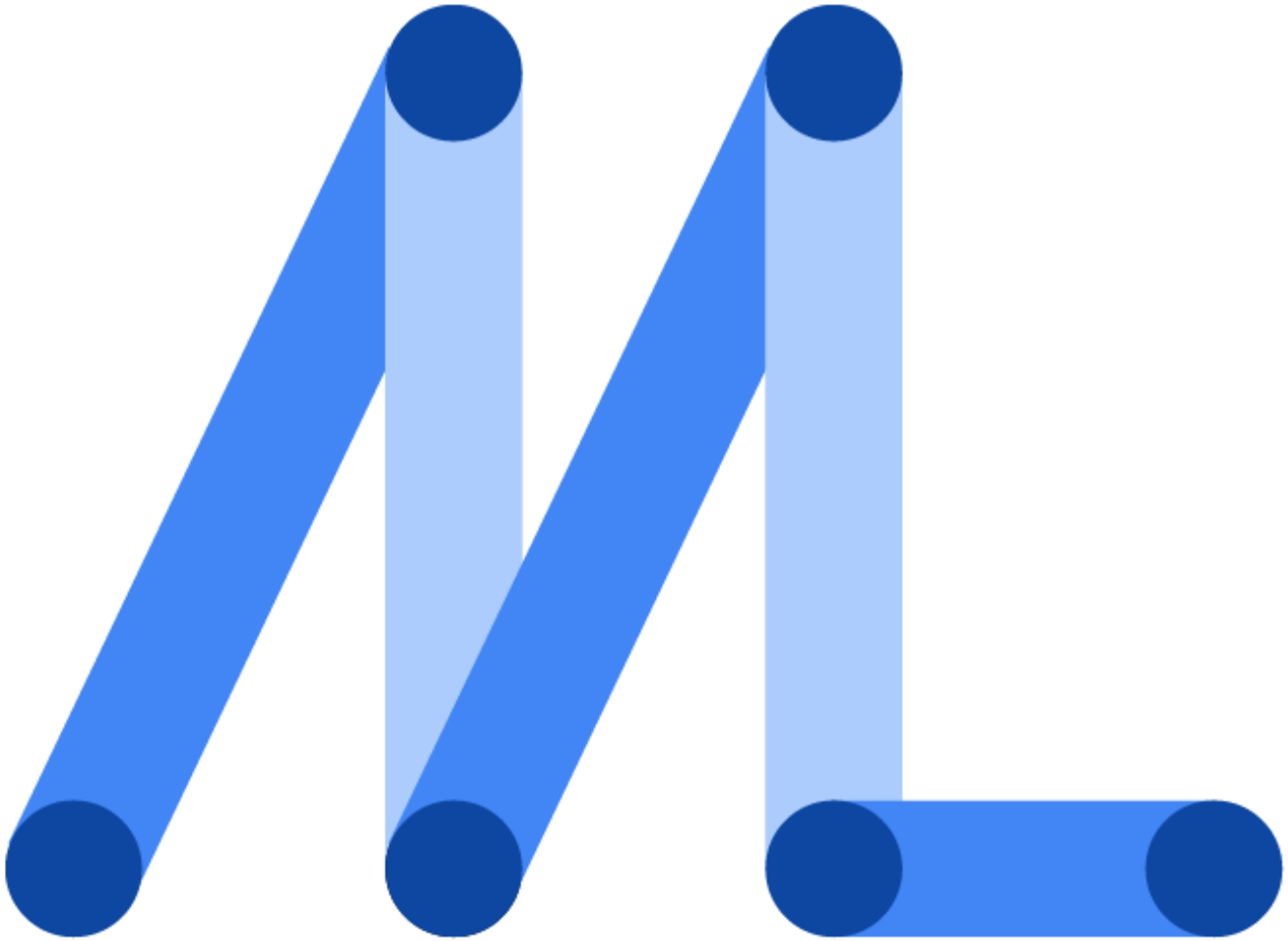
```
// Get the BluetoothDevice object using its MAC address
val device = bluetoothAdapter.getRemoteDevice(address)

// Get the UUID for the service (some devices have multiple UUIDs)
val uuid = device.uuids[0].uuid

// Create an RFCOMM socket (similar to TCP socket) to connect to the device
val socket = device.createRfcommSocketToServiceRecord(uuid)

// Connect to the remote device via the socket (must be on a background thread)
socket.connect()
```

Sending and Receiving Data

```
// Get the output stream of the connected socket
val outputStream: OutputStream = socket.outputStream

// Send data over the connection
outputStream.write(myByteArray)

// Get the input stream to receive data
val inputStream: InputStream = socket.inputStream

// Prepare a byte array buffer to hold incoming data
val bytes = ByteArray(1024)

// Read data from the input stream
val numBytes = inputStream.read(bytes)
// numBytes will tell you how many bytes were actually read
```

We will do an ICTE on Bluetooth Next week

ML Kit



What is ML Kit?

- Definition:
  - A mobile SDK by Google that enables machine learning features on Android and iOS
  - Provides ready-to-use solutions without needing ML expertise

- Two Types of Models:
  - On-device: Fast, private, works offline
  - Cloud-based: More powerful, needs internet

Available APIs in ML Kit

- Face Detection
- Text Recognition (OCR)
- Barcode Scanning
- Image Labeling
- Pose Detection
- Translation
- Smart Reply

Let’s use ML Kit in an example

Example Description

- Using the Drawing app we built last week, lets add an ML Kit with text recognition to detect the letter the user draws
- Flow
  - User draws a number or letter.
  - App captures the drawing.
  - App detects the text/number drawn!

How ML Kit Text Recognition Works

Flow:

- Capture image or drawing.
- Wrap it into an InputImage object.
- Feed it into ML Kit’s recognizer.
- Get back detected text!

Adding ML Kit Dependency

- In the app's build.gradle:
- implementation 'com.google.mlkit:text-recognition:16.0.0'

Follow these steps:

- In your DrawingView.kt, add a function:

```
fun getBitmap(): Bitmap {
    // Create a bitmap with same dimensions as view
    val bitmap = Bitmap.createBitmap(width, height,
Bitmap.Config.ARGB_8888)
    val canvas = Canvas(bitmap)
    draw(canvas) // Draw current canvas content onto bitmap
    return bitmap
}
```

What does this function do? Write it as a comment

Running ML Kit Text Recognition

After user finishes drawing (e.g., button click):

```
val recognizer = TextRecognition.getClient()
val inputImage = InputImage.fromBitmap(drawingView.getBitmap(), 0)

recognizer.process(inputImage)
    .addOnSuccessListener { visionText ->
        val detectedText = visionText.text
        // TODO: Show detected text on screen!
    }
    .addOnFailureListener { e ->
        // TODO: Show error message
    }
```

Displaying Recognized Text

- Example:
  - Toast the recognized text:

```
Toast.makeText(this, detectedText, Toast.LENGTH_LONG).show()
```

- Or update a TextView on the screen dynamically.

Put it all together

- Ask for help