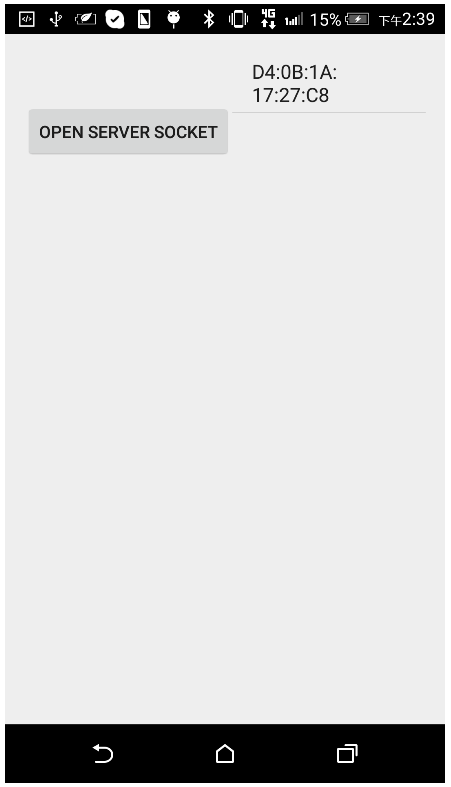
Bluetooth & Socket

Sample Code

- https://dl.dropboxusercontent.com/u/21274694/android/BluetoothExample.zip
- ListView of paired devices
- Toast of discovered devices
- A button to open server connection listener
- Send "Hello World" when you click the paired device (you can see the message in Log.i)



Setting up Bluetooth

- Always check the permission first

```
<manifest ...>
   <uses-permission android:name="android.permission.BLUETOOTH"  />
   ...
</manifest>
```

Make Sure the Device Supports BT

```
BluetoothAdapter mBluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
if (mBluetoothAdapter == null) {
    // Device does not support Bluetooth
}
```

Enable Your Bluetooth

```
BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

if (mBluetoothAdapter == null) {

  // Device does not support Bluetooth

}

else { // Device supports Bluetooth

     if (!mBluetoothAdapter.isEnabled()) {

        Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);

     }

}
```

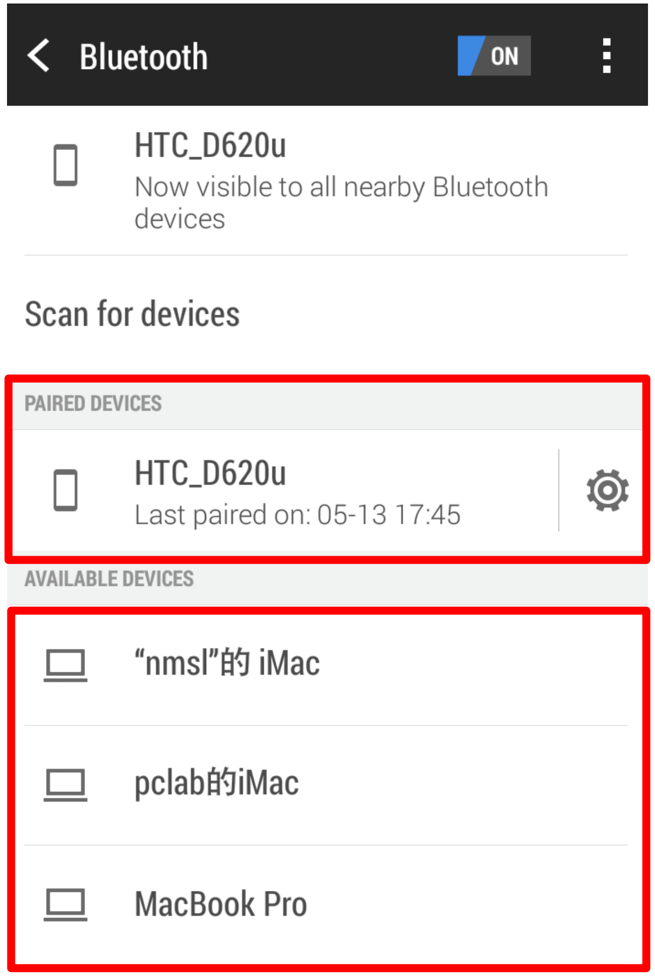
Query Paired Devices

- Before device discovery, you may want to see if the desired device is already known
- To do so, call getBondedDevices(). This will return a set of BluetoothDevices representing paired devices

```
Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
if (pairedDevices.size() > 0) { // If there are paired devices
   for (BluetoothDevice device : pairedDevices) {  // Loop through paired devices
      Toast.makeText(getApplicationContext(),  device.getName() + " Paired \n" +
device.getAddress(),  Toast.LENGTH_LONG).show();
   }
}
```

Discovering (1/2)

- desired devices is not paired → discovering
- We need to use Intent filter with Action = "BluetoothDevice.ACTION_FOUND"



HTC_D620u
Now visible to all nearby Bluetooth devices

Scan for devices

PAIRED DEVICES

HTC_D620u
Last paired on: 05-13 17:45

AVAILABLE DEVICES

"nmsl"的 iMac

pclab的iMac

MacBook Pro

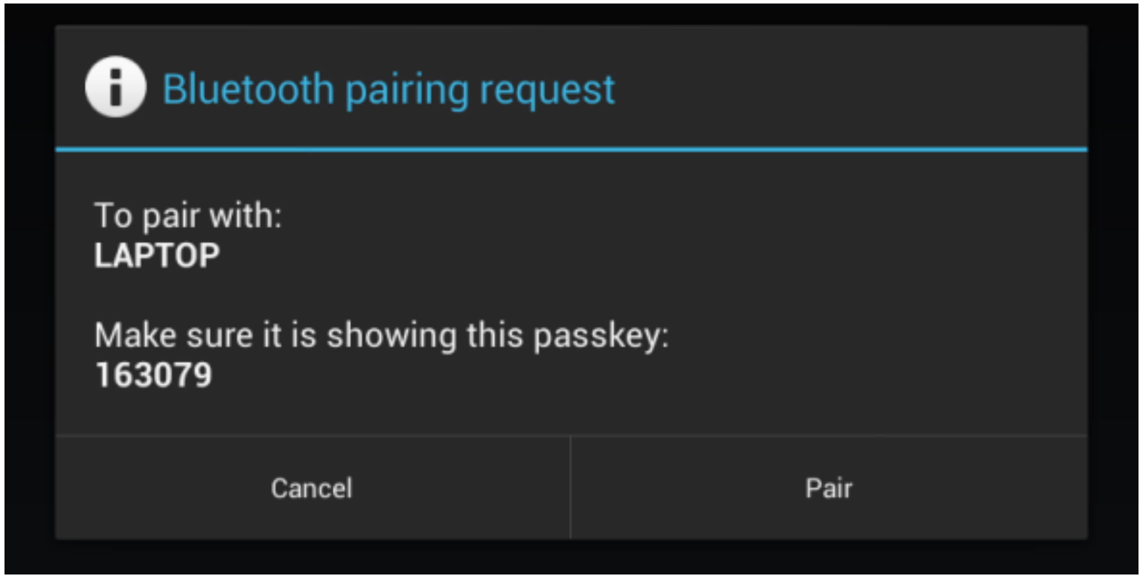
Discovering (2/2)

- After new an IntentFilter, we need to register a broacastReceiver to receive the information of other bluetooth devices

```
registerReceiver(mReceiver, filter);
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        // When discovery finds a device
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            // Get the BluetoothDevice object from the Intent
            BluetoothDevice device=intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            Toast.makeText(getApplicationContext(),device.getAddress(),
Toast.LENGTH_SHORT).show();
        }
    }
  };
```

What is Pairing?

- Two devices are aware of each other's existence, have a shared link-key that can be used for authentication


Bluetooth pairing request

To pair with:
LAPTOP

Make sure it is showing this passkey:
163079

Pairing

- manually
- use UUID to do socket connection and the Android system will automatically do pairing

Difference Between Pair and Connect

- Pair: two devices are known each other
- Connect: devices currently share an channel and are able to transmit data with each other
    - Devices are paired: directly connect to the other device
    - Devices are not paired: the android system will automatically do pairing and then connect to the other device

Enable Discoverability

- To make other devices can see you, enable your discoverability using Intent

```
public void enableDiscoverability(){
	Intent discoverableIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
	discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION,
300);
	startActivity(discoverableIntent);
  }
```

Connection - Server

- Create a thread to wait for connection
  - the function “**AcceptThread**” in the sample code

Connection – Server (code)

```
while (true) {
            try {
               Log.i("Server:", "Waiting for connection");
               socket = mmServerSocket.accept();
            } catch (IOException e) {
               break;
            }
            // If a connection was accepted, new a thread to receive the message "Hello
World" sent from client
            if (socket != null) {
               if (mConnectedThread != null) {mConnectedThread.cancel();
mConnectedThread = null;}
               mConnectedThread = new ConnectedThread(socket);
               mConnectedThread.start();
               Log.i("Server:", "one client connected");
            }
         }
```

Bluetooth Socket – read (server)

- the function “**ConnectedThread**” in your sample code

```
try {
            // Read from the InputStream
            bytes = mmInStream.read(buffer);
            String str = new String(buffer);
            Log.i("Server get message:", str);
        } catch (IOException e) {
            break;
        }
}
```

Bluetooth Socket - Client

- The function “**ConnectThread**” in sample code

```
mmSocket=device.createRfcommSocketToServiceRecord(MY_UUID);
mmSocket.connect();
  Log.i("Client:", "connect socket success");
```

bluetooth device in server side

Bluetooth Socket – write (client)

- the function “**ConnectThread**” in your sample code

```
String message="Hello World!";
mConnectedThread.write(message.getBytes());
public void write(byte[] bytes) {
        try {
            mmOutStream.write(bytes);
        } catch (IOException e) { }
}
```