**ภาคผนวก ง**

โปรแกรมการทำงานฟลัทเทอร์

โปรแกรมการทำงานฟลัทเทอร์ หน้าหลัก

|  |  |
| --- | --- |
| 1 | import 'package:digital\_lock\_v1\_flutter/SQLite/sqlite\_helpper.dart'; |
| 2 | import 'package:flutter/material.dart'; |
| 3 | import 'package:flutter\_blue/flutter\_blue.dart'; |
| 4 | import 'dart:async'; |
| 5 | import 'dart:convert' show utf8; |
| 6 | import 'package:wakelock/wakelock.dart'; |
| 7 | import 'package:google\_maps\_flutter/google\_maps\_flutter.dart'; |
| 8 | import 'package:location/location.dart'; |
| 9 |  |
| 10 | void main() => runApp(MyApp()); |
| 11 |  |
| 12 | class MyApp extends StatelessWidget { |
| 13 | @override |
| 14 | Widget build(BuildContext context) { |
| 15 | return MaterialApp( |
| 16 | title: 'RMUTR Digital Lock V1', |
| 17 | home: DigitalLock(), |
| 18 | ); |
| 19 | } |
| 20 | } |
| 21 |  |
| 22 | //----------------------------------------------------------------------------- |
| 23 | class DigitalLock extends StatefulWidget { |
| 24 | @override |
| 25 | \_DigitalLockState createState() => \_DigitalLockState(); |
| 26 | } |
| 27 |  |
| 28 | class \_DigitalLockState extends State<DigitalLock> { |
| 29 | Completer<GoogleMapController> \_controller = Completer(); |
| 30 | LatLng \_center = const LatLng(13.7181689, 100.4968919); |
| 31 | double latlngLat; |
| 32 | double latlngLng; |
| 33 | Marker marker; |
| 34 | String tx\_buff\_str = ""; |
| 35 | TextEditingController txtCtrl\_Pincode = TextEditingController(); |
| 36 | TextEditingController txtCtrl\_AlarmTime = TextEditingController(); |
| 37 | //----------------------------------------------------------------------------- |
| 38 | final String SERVICE\_UUID = "6e400001-b5a3-f393-e0a9-e50e24dcca9e"; |
| 39 | final String CHARACTERISTIC\_UUID = "6e400002-b5a3-f393-e0a9-e50e24dcca9e"; |
| 40 | final String TARGET\_DEVICE\_NAME = "Digital Lock V1"; |
| 41 |  |
| 42 | FlutterBlue flutterBlue = FlutterBlue.instance; |
| 43 | StreamSubscription<ScanResult> scanSubScription; |
| 44 |  |
| 45 | BluetoothDevice targetDevice; |
| 46 | BluetoothCharacteristic targetCharacteristic; |
| 47 |  |
| 48 | String connectionText = ""; |
| 49 |  |
| 50 | @override |
| 51 | void initState() { |
| 52 | super.initState(); |
| 53 | Wakelock.enable(); |
| 54 | startScan(); |
| 55 | readMap(); |
| 56 | } |
| 57 |  |
| 58 | startScan() { |
| 59 | setState(() { |
| 60 | connectionText = "01 Start Scanning"; |
| 61 | print(connectionText); |
| 62 | }); |
| 63 |  |
| 64 | scanSubScription = flutterBlue.scan().listen((scanResult) { |
| 65 | if (scanResult.device.name == TARGET\_DEVICE\_NAME) { |
| 66 | stopScan(); |
| 67 | setState(() { |
| 68 | connectionText = "02 Found Target Device"; |
| 69 | print(connectionText); |
| 70 | }); |
| 71 |  |
| 72 | targetDevice = scanResult.device; |
| 73 | connectToDevice(); |
| 74 | } |
| 75 | }, onDone: () => stopScan()); |
| 76 | } |
| 77 |  |
| 78 | stopScan() { |
| 79 | scanSubScription?.cancel(); |
| 80 | scanSubScription = null; |
| 81 | } |
| 82 |  |
| 83 | connectToDevice() async { |
| 84 | if (targetDevice == null) return; |
| 85 |  |
| 86 | setState(() { |
| 87 | connectionText = "03 Device Connecting"; |
| 88 | print(connectionText); |
| 89 | }); |
| 90 |  |
| 91 | await targetDevice.connect(); |
| 92 | setState(() { |
| 93 | connectionText = "04 Device Connected"; |
| 94 | print(connectionText); |
| 95 | }); |
| 96 |  |
| 97 | discoverServices(); |
| 98 | } |
| 99 |  |
| 100 | discoverServices() async { |
| 101 | if (targetDevice == null) return; |
| 102 |  |
| 103 | List<BluetoothService> services = await targetDevice.discoverServices(); |
| 104 | services.forEach((service) { |
| 105 | if (service.uuid.toString() == SERVICE\_UUID) { |
| 106 | service.characteristics.forEach((characteristic) { |
| 107 | if (characteristic.uuid.toString() == CHARACTERISTIC\_UUID) { |
| 108 | targetCharacteristic = characteristic; |
| 109 | writeData("Hi, ESP32"); |
| 110 | setState(() { |
| 111 | connectionText = "05 All Ready with ${targetDevice.name}"; |
| 112 | print(connectionText); |
| 113 | }); |
| 114 | } |
| 115 | }); |
| 116 | } |
| 117 | }); |
| 118 | } |
| 119 |  |
| 120 | disconnectFromDevice() { |
| 121 | if (targetDevice == null) return; |
| 122 |  |
| 123 | targetDevice.disconnect(); |
| 124 |  |
| 125 | setState(() { |
| 126 | connectionText = "06 Device Disconnected"; |
| 127 | print(connectionText); |
| 128 | }); |
| 129 | } |
| 130 |  |
| 131 | writeData(String data) { |
| 132 | if (targetCharacteristic == null) return; |
| 133 |  |
| 134 | List<int> bytes = utf8.encode(data); |
| 135 | targetCharacteristic.write(bytes); |
| 136 | } |
| 137 |  |
| 138 | //----------------------------------------------------------------------------- |
| 139 | void keepinTxBuff(String pstr) { |
| 140 | if (tx\_buff\_str.length < 10) { |
| 141 | tx\_buff\_str += pstr; |
| 142 | } |
| 143 | } |
| 144 |  |
| 145 | void sendTxBuff(String pstr) { |
| 146 | print(pstr); |
| 147 | writeData(pstr); |
| 148 | } |
| 149 |  |
| 150 | Future<void> changePincodeDialog() async { |
| 151 | this.txtCtrl\_Pincode.text = ""; |
| 152 |  |
| 153 | showDialog( |
| 154 | context: context, |
| 155 | builder: (value) => AlertDialog( |
| 156 | title: Text("New Pincode"), |
| 157 | content: TextField(controller: txtCtrl\_Pincode), |
| 158 | actions: <Widget>[ |
| 159 | FlatButton( |
| 160 | onPressed: () { |
| 161 | this.sendTxBuff( |
| 162 | "C3-" + tx\_buff\_str + "-" + txtCtrl\_Pincode.text); |
| 163 | Navigator.of(context).pop(); |
| 164 | }, |
| 165 | child: Text("Ok")), |
| 166 | FlatButton( |
| 167 | onPressed: () => Navigator.of(context).pop(), |
| 168 | child: Text("Cancel")), |
| 169 | ], |
| 170 | ), |
| 171 | ); |
| 172 | } |
| 173 |  |
| 174 | Future<void> setAlarmDialog() async { |
| 175 | this.txtCtrl\_AlarmTime.text = ""; |
| 176 |  |
| 177 | showDialog( |
| 178 | context: context, |
| 179 | builder: (value) => AlertDialog( |
| 180 | title: Text("Time to alarm (sec.)"), |
| 181 | content: TextField(controller: txtCtrl\_AlarmTime), |
| 182 | actions: <Widget>[ |
| 183 | FlatButton( |
| 184 | onPressed: () { |
| 185 | String btime\_str = txtCtrl\_AlarmTime.text; |
| 186 | int btime\_int = int.tryParse(btime\_str) ?? 0; |
| 187 | btime\_str = btime\_int.toString().padLeft(4, '0'); |
| 188 | this.sendTxBuff("C4-" + tx\_buff\_str + "-" + btime\_str); |
| 189 | Navigator.of(context).pop(); |
| 190 | }, |
| 191 | child: Text("Ok")), |
| 192 | FlatButton( |
| 193 | onPressed: () => Navigator.of(context).pop(), |
| 194 | child: Text("Cancel")), |
| 195 | ], |
| 196 | ), |
| 197 | ); |
| 198 | } |
| 199 |  |
| 200 | Widget dlBtn\_0() { |
| 201 | return Container( |
| 202 | width: 80.0, |
| 203 | child: RaisedButton( |
| 204 | child: Text("0"), |
| 205 | onPressed: () => this.keepinTxBuff("0"), |
| 206 | ), |
| 207 | ); |
| 208 | } |
| 209 |  |
| 210 | Widget dlBtn\_1() { |
| 211 | return Container( |
| 212 | width: 80.0, |
| 213 | child: RaisedButton( |
| 214 | child: Text("1"), |
| 215 | onPressed: () => this.keepinTxBuff("1"), |
| 216 | ), |
| 217 | ); |
| 218 | } |
| 219 |  |
| 220 | Widget dlBtn\_2() { |
| 221 | return Container( |
| 222 | width: 80.0, |
| 223 | child: RaisedButton( |
| 224 | child: Text("2"), |
| 225 | onPressed: () => this.keepinTxBuff("2"), |
| 226 | ), |
| 227 | ); |
| 228 | } |
| 229 |  |
| 230 | Widget dlBtn\_3() { |
| 231 | return Container( |
| 232 | width: 80.0, |
| 233 | child: RaisedButton( |
| 234 | child: Text("3"), |
| 235 | onPressed: () => this.keepinTxBuff("3"), |
| 236 | ), |
| 237 | ); |
| 238 | } |
| 239 |  |
| 240 | Widget dlBtn\_4() { |
| 241 | return Container( |
| 242 | width: 80.0, |
| 243 | child: RaisedButton( |
| 244 | child: Text("4"), |
| 245 | onPressed: () => this.keepinTxBuff("4"), |
| 246 | ), |
| 247 | ); |
| 248 | } |
| 249 |  |
| 250 | Widget dlBtn\_5() { |
| 251 | return Container( |
| 252 | width: 80.0, |
| 253 | child: RaisedButton( |
| 254 | child: Text("5"), |
| 255 | onPressed: () => this.keepinTxBuff("5"), |
| 256 | ), |
| 257 | ); |
| 258 | } |
| 259 |  |
| 260 | Widget dlBtn\_6() { |
| 261 | return Container( |
| 262 | width: 80.0, |
| 263 | child: RaisedButton( |
| 264 | child: Text("6"), |
| 265 | onPressed: () => this.keepinTxBuff("6"), |
| 266 | ), |
| 267 | ); |
| 268 | } |
| 269 |  |
| 270 | Widget dlBtn\_7() { |
| 271 | return Container( |
| 272 | width: 80.0, |
| 273 | child: RaisedButton( |
| 274 | child: Text("7"), |
| 275 | onPressed: () => this.keepinTxBuff("7"), |
| 276 | ), |
| 277 | ); |
| 278 | } |
| 279 |  |
| 280 | Widget dlBtn\_8() { |
| 281 | return Container( |
| 282 | width: 80.0, |
| 283 | child: RaisedButton( |
| 284 | child: Text("8"), |
| 285 | onPressed: () => this.keepinTxBuff("8"), |
| 286 | ), |
| 287 | ); |
| 288 | } |
| 289 |  |
| 290 | Widget dlBtn\_9() { |
| 291 | return Container( |
| 292 | width: 80.0, |
| 293 | child: RaisedButton( |
| 294 | child: Text("9"), |
| 295 | onPressed: () => this.keepinTxBuff("9"), |
| 296 | ), |
| 297 | ); |
| 298 | } |
| 299 |  |
| 300 | Widget dlBtnCancel() { |
| 301 | return Container( |
| 302 | width: 80.0, |
| 303 | child: RaisedButton( |
| 304 | child: Text("Cancel"), |
| 305 | onPressed: () => tx\_buff\_str = "", |
| 306 | ), |
| 307 | ); |
| 308 | } |
| 309 |  |
| 310 | Widget dlBtnOk() { |
| 311 | return Container( |
| 312 | width: 80.0, |
| 313 | child: RaisedButton( |
| 314 | child: Text("Ok"), |
| 315 | onPressed: () { |
| 316 | print(connectionText); |
| 317 | this.sendTxBuff("C2-" + tx\_buff\_str); |
| 318 | }), |
| 319 | ); |
| 320 | } |
| 321 |  |
| 322 | Widget dlBtnStop() { |
| 323 | return Container( |
| 324 | width: 100.0, |
| 325 | child: RaisedButton( |
| 326 | color: Colors.red, |
| 327 | child: Text("Stop"), |
| 328 | onPressed: () { |
| 329 | this.sendTxBuff("C0-" + tx\_buff\_str); |
| 330 | if (targetDevice != null) { |
| 331 | \_initGps(); |
| 332 | } |
| 333 | }), |
| 334 | ); |
| 335 |  |
| 336 | } |
| 337 |  |
| 338 | Widget dlBtnStart() { |
| 339 | return Container( |
| 340 | width: 100.0, |
| 341 | child: RaisedButton( |
| 342 | color: Colors.green, |
| 343 | child: Text("Start"), |
| 344 | onPressed: () => this.sendTxBuff("C1-" + tx\_buff\_str), |
| 345 | ), |
| 346 | ); |
| 347 | } |
| 348 |  |
| 349 | Widget dlBtnChangePincode() { |
| 350 | return Container( |
| 351 | width: 100.0, |
| 352 | child: RaisedButton( |
| 353 | color: Colors.yellow[300], |
| 354 | child: Text("Change Pincode"), |
| 355 | onPressed: () => this.changePincodeDialog(), |
| 356 | ), |
| 357 | ); |
| 358 | } |
| 359 |  |
| 360 | Widget dlBtnAlarm() { |
| 361 | return Container( |
| 362 | width: 100.0, |
| 363 | child: OutlineButton( |
| 364 | child: Text("Alarm"), |
| 365 | onPressed: () => this.setAlarmDialog(), |
| 366 | ), |
| 367 | ); |
| 368 | } |
| 369 |  |
| 370 | Widget dlKeyPad() { |
| 371 | return ListView( |
| 372 | children: <Widget>[ |
| 373 | Container( |
| 374 | height: 20, |
| 375 | ), |
| 376 | Row( |
| 377 | mainAxisAlignment: MainAxisAlignment.spaceEvenly, |
| 378 | children: <Widget>[ |
| 379 | dlBtn\_1(), |
| 380 | dlBtn\_2(), |
| 381 | dlBtn\_3(), |
| 382 | dlBtnStop(), |
| 383 | ], |
| 384 | ), |
| 385 | Row( |
| 386 | mainAxisAlignment: MainAxisAlignment.spaceEvenly, |
| 387 | children: <Widget>[ |
| 388 | dlBtn\_4(), |
| 389 | dlBtn\_5(), |
| 390 | dlBtn\_6(), |
| 391 | dlBtnStart(), |
| 392 | ], |
| 393 | ), |
| 394 | Row( |
| 395 | mainAxisAlignment: MainAxisAlignment.spaceEvenly, |
| 396 | children: <Widget>[ |
| 397 | dlBtn\_7(), |
| 398 | dlBtn\_8(), |
| 399 | dlBtn\_9(), |
| 400 | dlBtnChangePincode(), |
| 401 | ], |
| 402 | ), |
| 403 | Row( |
| 404 | mainAxisAlignment: MainAxisAlignment.spaceEvenly, |
| 405 | children: <Widget>[ |
| 406 | dlBtnCancel(), |
| 407 | dlBtn\_0(), |
| 408 | dlBtnOk(), |
| 409 | dlBtnAlarm(), |
| 410 | ], |
| 411 | ), |
| 412 | Container( |
| 413 | child: showLogo(), |
| 414 | ) |
| 415 | ], |
| 416 | ); |
| 417 | } |
| 418 |  |
| 419 | List<Widget> mapButtom(BuildContext context) { |
| 420 | return <Widget>[ |
| 421 | IconButton( |
| 422 | icon: Icon(Icons.map), |
| 423 | color: Colors.blue[100], |
| 424 | iconSize: 35, |
| 425 | onPressed: () { |
| 426 | print('buttom refresh'); |
| 427 | setState(() { |
| 428 | MaterialPageRoute materialPageRoute = MaterialPageRoute( |
| 429 | builder: (BuildContext context) => googleMap(), |
| 430 | ); |
| 431 | Navigator.of(context).push(materialPageRoute); |
| 432 | \_onAddMarkerButtonPressed(); |
| 433 | }); |
| 434 | }) |
| 435 | ]; |
| 436 | } |
| 437 |  |
| 438 | Future readMap() async { |
| 439 | var object = await SQLiteHelper().readAllDataFromSQLite(); |
| 440 | latlngLat = double.parse(object[0].latSqlite); |
| 441 | latlngLng = double.parse(object[0].lonSqlite); |
| 442 | \_center = LatLng(latlngLat, latlngLng); |
| 443 | print('lat = ${object[0].latSqlite} lon = ${object[0].lonSqlite}'); |
| 444 | } |
| 445 |  |
| 446 | Future<LocationData> findLocationData() async { |
| 447 | Location location = Location(); |
| 448 | try { |
| 449 | return location.getLocation(); |
| 450 | } catch (e) { |
| 451 | return null; |
| 452 | } |
| 453 | } |
| 454 |  |
| 455 | Future \_goToTheLake() async { |
| 456 | final GoogleMapController controller = await \_controller.future; |
| 457 | controller.animateCamera( |
| 458 | CameraUpdate.newCameraPosition( |
| 459 | CameraPosition( |
| 460 | target: \_center, |
| 461 | zoom: 19, |
| 462 | tilt: 30, //bearing: 192.8334901395799, |
| 463 | ), |
| 464 | ), |
| 465 | ); |
| 466 | } |
| 467 |  |
| 468 | Widget googleMap() { |
| 469 | return Scaffold( |
| 470 | appBar: AppBar( |
| 471 | backgroundColor: Colors.orange, |
| 472 | title: Text("last parking location"), |
| 473 | ), |
| 474 | body: GoogleMap( |
| 475 | initialCameraPosition: CameraPosition( |
| 476 | target: \_center, |
| 477 | zoom: 15, |
| 478 | tilt: 30, |
| 479 | ), |
| 480 | markers: Set.of((marker != null) ? [marker] : []), |
| 481 | onMapCreated: (GoogleMapController controller) { |
| 482 | \_controller.complete(controller); |
| 483 | }, |
| 484 | ), |
| 485 | ); |
| 486 | } |
| 487 |  |
| 488 | Widget showLogo() { |
| 489 | return Image.asset('images/logo.png'); |
| 490 | } |
| 491 |  |
| 492 | \_initGps() async { |
| 493 | LocationData locationData = await findLocationData(); |
| 494 | latlngLat = locationData.latitude; |
| 495 | latlngLng = locationData.longitude; |
| 496 | setState(() { |
| 497 | \_center = LatLng(latlngLat, latlngLng); |
| 498 | \_goToTheLake(); |
| 499 | \_onAddMarkerButtonPressed(); |
| 500 | addMap(); |
| 501 | }); |
| 502 | } |
| 503 |  |
| 504 | addMap() async { |
| 505 | await SQLiteHelper().deleteDataWhereId(1).then((value) {}); |
| 506 |  |
| 507 | Map<String, dynamic> map = Map(); |
| 508 |  |
| 509 | map['latSqlite'] = latlngLat.toString(); |
| 510 | map['lonSqlite'] = latlngLng.toString(); |
| 511 | map['txSqlute'] = tx\_buff\_str; |
| 512 |  |
| 513 | print('map ==> ${map.toString()}'); |
| 514 |  |
| 515 | CartModel cartModel = CartModel.fromJson(map); |
| 516 |  |
| 517 | await SQLiteHelper().insertDataToSQLite(cartModel).then((value) { |
| 518 | print('Insert Success'); |
| 519 | }); |
| 520 | } |
| 521 |  |
| 522 | \_onAddMarkerButtonPressed() { |
| 523 | setState(() { |
| 524 | marker = Marker( |
| 525 | markerId: MarkerId("home"), |
| 526 | position: \_center, |
| 527 | infoWindow: InfoWindow( |
| 528 | title: 'This is a Title', |
| 529 | snippet: 'This is a snippet', |
| 530 | ), |
| 531 | draggable: false, |
| 532 | flat: true, |
| 533 | icon: BitmapDescriptor.defaultMarker, |
| 534 | ); |
| 535 | }); |
| 536 | } |
| 537 |  |
| 538 | @override |
| 539 | Widget build(BuildContext context) { |
| 540 | return Scaffold( |
| 541 | appBar: AppBar( |
| 542 | backgroundColor: Colors.orange, |
| 543 | title: Text("RMUTR Digital Lock V1"), |
| 544 | actions: mapButtom(context), |
| 545 | ), |
| 546 | body: dlKeyPad(), |
| 547 | ); |
| 548 | } |
| 549 | } |
| 550 |  |
| 551 | class CartModel { |
| 552 | int id; |
| 553 | String latSqlite; |
| 554 | String lonSqlite; |
| 555 |  |
| 556 | CartModel({this.id, this.latSqlite, this.lonSqlite}); |
| 557 |  |
| 558 | CartModel.fromJson(Map<String, dynamic> json) { |
| 559 | id = json['id']; |
| 560 | latSqlite = json['latSqlite']; |
| 561 | lonSqlite = json['lonSqlite']; |
| 562 | } |
| 563 |  |
| 564 | Map<String, dynamic> toJson() { |
| 565 | final Map<String, dynamic> data = new Map<String, dynamic>(); |
| 566 | data['id'] = this.id; |
| 567 | data['latSqlite'] = this.latSqlite; |
| 568 | data['lonSqlite'] = this.lonSqlite; |
| 569 | return data; |
| 570 | } |
| 571 | } |

โปรแกรมการทำงานฟลัทเทอร์ หน้าเอสคิวไลท์

|  |  |
| --- | --- |
| 1 | import 'package:digital\_lock\_v1\_flutter/SQLite/sqlite\_helpper.dart'; |
| 2 | import 'package:flutter/material.dart'; |
| 3 | import 'package:flutter\_blue/flutter\_blue.dart'; |
| 4 | import 'dart:async'; |
| 5 | import 'dart:convert' show utf8; |
| 6 | import 'package:wakelock/wakelock.dart'; |
| 7 | import 'package:google\_maps\_flutter/google\_maps\_flutter.dart'; |
| 8 | import 'package:location/location.dart'; |
| 9 |  |
| 10 | void main() => runApp(MyApp()); |
| 11 |  |
| 12 | class MyApp extends StatelessWidget { |
| 13 | @override |
| 14 | Widget build(BuildContext context) { |
| 15 | return MaterialApp( |
| 16 | title: 'RMUTR Digital Lock V1', |
| 17 | home: DigitalLock(), |
| 18 | ); |
| 19 | } |
| 20 | } |
| 21 |  |
| 22 | //----------------------------------------------------------------------------- |
| 23 | class DigitalLock extends StatefulWidget { |
| 24 | @override |
| 25 | \_DigitalLockState createState() => \_DigitalLockState(); |
| 26 | } |
| 27 |  |
| 28 | class \_DigitalLockState extends State<DigitalLock> { |
| 29 | Completer<GoogleMapController> \_controller = Completer(); |
| 30 | LatLng \_center = const LatLng(13.7181689, 100.4968919); |
| 31 | double latlngLat; |
| 32 | double latlngLng; |
| 33 | Marker marker; |
| 34 | String tx\_buff\_str = ""; |
| 35 | TextEditingController txtCtrl\_Pincode = TextEditingController(); |
| 36 | TextEditingController txtCtrl\_AlarmTime = TextEditingController(); |
| 37 | //----------------------------------------------------------------------------- |
| 38 | final String SERVICE\_UUID = "6e400001-b5a3-f393-e0a9-e50e24dcca9e"; |
| 39 | final String CHARACTERISTIC\_UUID = "6e400002-b5a3-f393-e0a9-e50e24dcca9e"; |
| 40 | final String TARGET\_DEVICE\_NAME = "Digital Lock V1"; |
| 41 |  |
| 42 | FlutterBlue flutterBlue = FlutterBlue.instance; |
| 43 | StreamSubscription<ScanResult> scanSubScription; |
| 44 |  |
| 45 | BluetoothDevice targetDevice; |
| 46 | BluetoothCharacteristic targetCharacteristic; |
| 47 |  |
| 48 | String connectionText = ""; |
| 49 |  |
| 50 | @override |
| 51 | void initState() { |
| 52 | super.initState(); |
| 53 | Wakelock.enable(); |
| 54 | startScan(); |
| 55 | readMap(); |
| 56 | } |
| 57 |  |
| 58 | startScan() { |
| 59 | setState(() { |
| 60 | connectionText = "01 Start Scanning"; |
| 61 | print(connectionText); |
| 62 | }); |