

М3 MOBILE

Руководство по Интеграции SDK WR15 (пакет средств разработки ПО)



V1.1

2025/11/13

Revision History

Date	Version	Comments	Author	Reviewer
2025/10/21	V1.0	Initial	Seongrak Choi	
2025/11/13	V1.1	Changed SDK distribution	Seongrak Choi	

Содержание

1. Импорт SDK	- 7 -
1.1 Библиотеки.....	- 7 -
2. Запрос разрешений.....	- 8 -
2.1 Если основной фокус на Android 12+ (API 31+).....	- 8 -
2.2 В случае, если необходимо поддерживать Android 6–11 (API 25–30).....	- 8 -
2.3 Для работы SDK также запрашивает определенные разрешения, которые являются обязательными (Runtime Permission Request Check).....	- 8 -
3. Инициализация SDK.....	- 9 -
3.1 Инициализация библиотек зависимостей.....	- 9 -
3.1.1 M3Wr15SdkConfig	- 9 -
3.1.2 Context	- 9 -
4. Поиск WR15.....	- 10 -
4.1 S2P (Scan to Pair)	- 10 -
4.1.1 Генерация Штрихкода.....	- 10 -
4.1.2 Start S2P Scan.....	- 10 -
4.1.3 Stop S2P Scan.....	- 10 -
4.2 Поиск и подключение.....	- 11 -
4.2.1 Start Discovery Scan	- 11 -
4.2.2 Stop Discovery Scan.....	- 12 -
5. Подключение к WR15	- 13 -
5.1 Connect.....	- 13 -
6. Разрыв Соединения с WR15	- 13 -
6.1 Disconnect to WR15.....	- 13 -
6.2 Observe Disconnect.....	- 14 -
7. Прием Отсканированных Данных.....	- 15 -
7.1 Add Barcode Data	- 15 -

7.2 Remove Barcode Data.....	- 15 -
8. Сведения об устройстве.....	- 16 -
8.1 Observe DeviceState	- 16 -
8.2 Set Scanner Setting	- 16 -
8.2.1 SettingCommand.....	- 16 -
8.2.2 Generate SettingCommand.....	- 16 -
8.2.3 Предостережение	- 18 -
9. DeviceState.....	- 19 -
9.1 Strucutre	- 19 -
9.1.1 DeviceInfo	- 19 -
9.1.2 GeneralSettings.....	- 19 -
9.1.3 Symbologies	- 19 -
9.1.4 BasicFormat.....	- 27 -
9.1.5 readerSettings.....	- 27 -
9.1.6 DevSettings.....	- 27 -
10. Settings.....	- 28 -
10.1 AustralianPostal	- 28 -
10.2 Aztec	- 28 -
10.3 Chinese2of5.....	- 28 -
10.4 Codabar	- 28 -
10.4.1 CodabarType.....	- 28 -
10.5 Code11	- 29 -
10.5.1 Code11Type	- 29 -
10.6 Code39	- 29 -
10.6.1 Code39Type	- 30 -
10.7 Code93	- 30 -
10.8 Code128	- 30 -

10.8.1 Code128Type.....	- 31 -
10.9 Composite.....	- 31 -
10.9.1 CompositeType	- 31 -
10.10 DataMatrix	- 31 -
10.10.1 DatatMatrixType.....	- 32 -
10.11 Discrete2Of5.....	- 32 -
10.12 DotCode	- 32 -
10.12.1 DotCodeType.....	- 32 -
10.13 EAN8.....	- 33 -
10.14 EAN13	- 33 -
10.15 Gs1DataBar14	- 33 -
10.16 Gs1DataBarExpanded	- 33 -
10.17 Gs1DataBarLimited	- 33 -
10.17.1 Gs1DataBarLimitedType.....	- 33 -
10.18 Gs1128	- 33 -
10.19 HanXin.....	- 34 -
10.20 Interleaved2Of5	- 34 -
10.20.1 Interleaved2Of5Type	- 34 -
10.21 ISBT128.....	- 35 -
10.22 JapanesePostal	- 35 -
10.23 Korean3of5.....	- 35 -
10.24 Matrix2of5.....	- 35 -
10.25 MaxiCode.....	- 35 -
10.26 MicroPdf417.....	- 35 -
10.27 MicroQrCode.....	- 36 -
10.28 MSI	- 36 -
10.28.1 MsiType	- 36 -

10.29 NetherlandsKix	- 36 -
10.30 Pdf417	- 36 -
10.31 QrCode.....	- 37 -
10.32 UKPostal.....	- 37 -
10.33 UPCA	- 37 -
10.33.1 UpcAType.....	- 37 -
10.34 UPCE1	- 37 -
10.34.1 UpcE1Type	- 38 -
10.35 UPCEan	- 38 -
10.35.1 UpcEanType	- 39 -
10.36 UPCE.....	- 39 -
10.36.1 UpcEType.....	- 40 -
10.37 UpuFicsPostal.....	- 40 -
10.38 USPlanet	- 40 -
10.39 USPostnet.....	- 40 -
10.40 General	- 40 -
10.40.1 SoundType	- 41 -
10.41 ReaderParams	- 41 -
10.41.1 ReaderType	- 41 -
10.42 BasicDataFormat	- 42 -
10.42.1 BasicFormatType.....	- 42 -
10.42.2 String Value Format.....	- 42 -
10.43 Dev	- 43 -
11. M3Utils	- 44 -

1. Импорт SDK

1.1 Библиотеки

1. Добавьте URL Maven-репозитория в файл **settings.gradle.kts**

```
1. dependencyResolutionManagement {  
2.     repositories {  
3.         google()  
4.         mavenCentral()  
5.  
6.         //Добавьте эту строку  
7.         maven("https://m3mobile.github.io/wr15-sdk-maven")  
8.     }  
9. }
```

* Если вы используете **Groovy**

```
1. dependencyResolutionManagement {  
2.     repositories {  
3.         google()  
4.         mavenCentral()  
5.  
6.         // Добавьте эту строку  
7.         maven { url "https://m3mobile.github.io/wr15-sdk-maven" }  
8.     }  
9.  
10. }
```

2. В файл **build.gradle (Module: app)** импортируйте SDK, указав соответствующую версию.

```
1. dependencies {  
2.     implementation("com.m3:wr15-sdk:1.2.0") //Добавьте соответствующую версию  
3.     ...  
4. }
```

2. Запрос разрешений

Для использования SDK обязательно требуются разрешения: Bluetooth и Геолокация (Местоположение). Зарегистрируйте соответствующие разрешения в зависимости от того, на какую версию Android ориентировано разрабатываемое приложение.

2.1 Если основной фокус на Android 12+ (API 31+)

1. <uses-permission android:name="android.permission.BLUETOOTH_SCAN"
2. android:usesPermissionFlags="neverForLocation"/>
3. <uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>

2.2 В случае, если необходимо поддерживать Android 6–11 (API 25–30)

1. <!-- Android 6~11 (API 25~30) -->
2. <uses-permission android:name="android.permission.BLUETOOTH"
3. android:maxSdkVersion="30"/>
4. <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
5. android:maxSdkVersion="30"/>
6. <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
7. android:maxSdkVersion="30"/>
8. <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
9. android:maxSdkVersion="30"/>
10. <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
11. android:maxSdkVersion="30"/>
- 12.
13. <!-- Android 12+ (API 31+) -->
14. <uses-permission android:name="android.permission.BLUETOOTH_SCAN"
15. android:usesPermissionFlags="neverForLocation"/>
16. <uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>

2.3 Для работы SDK также запрашивает определенные разрешения, которые являются обязательными (Runtime Permission Request Check).

- BLUETOOTH_SCAN
- BLUETOOTH_CONNECT
- ACCESS_FINE_LOCATION
- ACCESS_COARSE_LOCATION

Вышеупомянутые разрешения являются обязательными во время выполнения (Runtime Permissions), которые требуют непосредственного подтверждения от пользователя приложения,

пожалуйста, обратите внимание на статус разрешения Bluetooth перед использованием SDK.

3. Инициализация SDK

3.1 Инициализация библиотек зависимостей

Перед использованием библиотеки обязательно проведите инициализацию с помощью функции **initialize**.

Kotlin

```
1. val config = M3Wr15SdkConfig.builder()
2.     .connectionType(ConnectionType.SPP)
3.     .enableLog(true)
4.     .build()
5.
6.
7. M3Wr15Sdk.initialize(context, config)
8.
```

Java

```
1. M3Wr15SdkConfig config = M3Wr15SdkConfig.builder()
2.     .connectionType(ConnectionType.SPP)
3.     .enableLog(true)
4.     .build();
5.
6. M3Wr15Sdk sdk =
7.     M3Wr15Sdk.initialize(context, config);
```

3.1.1 M3Wr15SdkConfig

- *connectionType()* : Способ связи (коммуникации) с WR15

* ConnectionType.BLE не поддерживается в версии v1.0.

- *enableLog()* : Включение/выключение вывода логов SDK

3.1.2 Context

Поскольку Context используется для сканирования и работы с SharedPreferences, передавайте **Application Context**, а не **Activity Context**.

4. Поиск WR15

4.1 S2P (Scan to Pair)

Отсканируйте S2P штрихкод с помощью устройства WR15, чтобы найти отсканированное устройство.

4.1.1 Генерация Штрихкода

```
1. /* M3Wr15Sdk */
2.
3. @JvmOverloads
4. @JvmStatic
5. fun getBarcodeBitmapForS2P(width: Int = 260, height: Int = 50): Bitmap?
6.
7. @JvmStatic
8. fun getCmdForS2P(): String
```

Для генерации **обычного штрихкода(GS128??)**, вы можете реализовать **S2P**, используя **битовую карту (Bitmap)**, возвращаемую методом **getBarcodeBitmapForS2P()**. Однако, если вам нужен **другой тип штрихкода**, вы можете использовать **строковое значение (String)**, возвращаемое методом **getCmdForS2P()**, для создания и использования нужного вам типа штрихкода.

4.1.2 Start S2P Scan

В момент сканирования сгенерированного выше штрихкода, должно быть активно сканирование S2P (S2P Scan) для S2P.

For Kotlin

```
1. /* M3Wr15Sdk */
2.
3. @RequiresPermission(
4.     allOf = [Manifest.permission.BLUETOOTH_SCAN, Manifest.permission.BLUETOOTH_CONNECT])
5. fun startS2PScan(
6.     onDeviceFound: (BluetoothDevice) -> Unit,
7.     onScanFailed: (errorCode: Int) -> Unit
8. )
```

For Java

```
1. /* M3Wr15Sdk */
2.
3. @JvmStatic
4. @RequiresPermission(
5.     allOf = [Manifest.permission.BLUETOOTH_SCAN, Manifest.permission.BLUETOOTH_CONNECT])
6. fun startS2PScan(listener: S2PScanResultListener)
7.
8. interface S2PScanResultListener {
9.     fun onDeviceFound(device: BluetoothDevice)
10.    fun onScanFailed(errorCode: Int)
11. }
```

Для метода **startS2PScan()** обязательно требуется разрешение **BLUETOOTH_SCAN** и **BLUETOOTH_CONNECT**. Поэтому рекомендуется использовать **защитный код (defensive code)** для проверки наличия этих разрешений перед вызовом метода.

4.1.3 Stop S2P Scan

Сразу после обнаружения кольцевого сканера WR15 вы должны остановить startS2PScan(). Также желательно (или рекомендуется) остановить startS2PScan(), если вы покидаете экран, на котором присутствует S2P-штрихкод.

```

1. /* M3Wr15Sdk */
2.
3. @RequiresPermission(Manifest.permission.BLUETOOTH_SCAN)
4. fun stopS2PScan()
```

Example

```

1. @RequiresPermission(Manifest.permission.BLUETOOTH_SCAN)
2. fun stopS2PScan()
3.
4. M3Wr15Sdk.startS2PScan(object: S2PScanResultListener) {
5.     override fun onDeviceFound(device: BluetoothDevice) {
6.         M3Wr15Sdk.stopS2PScan() // Остановка сканирования
7.         // Попытка подключения (установления соединения) к обнаруженному BluetoothDevice
8.     }
9.
10.    override fun onScanFailed(errorCode: Int) {
11.        // Обработка ошибки сканирования (или Обработка сбоя сканирования)
12.    }
13. }
```

4.2 Поиск и подключение

Поиск (или Обнаружение) устройств-сканеров WR15 поблизости (в окружении)

4.2.1 Start Discovery Scan

Поскольку это работает как Discovery Scan (Поиск/Обнаружение), сканирование автоматически завершится примерно через 20 секунд после его начала.

```

1. /*M3Wr15Sdk*/
2.
3. @RequiresPermission(allOf = [Manifest.permission.BLUETOOTH_SCAN,
   Manifest.permission.BLUETOOTH_CONNECT])
4. fun startDiscoveryScan(listener: DiscoveryScanResultListener)
5.
6. interface DiscoveryScanResultListener {
7.     fun onStart() // Запускает сканирование
8.     fun onError() // Возникла ошибка
9.     fun onDeviceFound(device: DiscoveryDeviceModel) // Обнаружен сканер WR15
10.    fun onFinish() // Сканирование завершено
11. }
```

Exmaple

```

1. M3Wr15Sdk.startDiscoveryScan(object : DiscoveryScanResultListener {
2.     @RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
3.     override fun onDeviceFound(device: DiscoveryDeviceModel) {
4.         // Обработка обнаружения устройства.
5.     }
6.
7.     override fun onError(errorMsg: String) {
8.         // Обработка ошибок.
9.     }
10.
11.    override fun onFinish() {
```

```
12.    // Обработка ошибок.  
13. }  
14.  
15. override fun onStarteed() {  
16.    // Обработка начала сканирования.  
17. }  
18. }  
19. )
```

4.2.2 Stop Discovery Scan

Остановить Discovery Scan.

```
1. @JvmStatic  
2. fun stopDiscoveryScan()
```

5. Подключение к WR15

5.1 Connect

Подключить обнаруженное устройство **BluetoothDevice** к **SDK**.

For Kotlin

```
1. /*M3Wr15Sdk*/
2.
3. @RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
4. fun connectToDevice(
5.     device: BluetoothDevice,
6.     onSuccess: (session: TransportSession) -> Unit,
7.     onFailure: (errorMsg: String) -> Unit
8. )
```

For Java

```
1. /*M3Wr15Sdk*/
2.
3. @JvmStatic
4. @RequiresPermission(Manifest.permission.BLUETOOTH_CONNECT)
5. fun connectToDevice(device: BluetoothDevice, listener: ConnectResultListener)
6.
7. interface ConnectResultListener {
8.     fun onSuccess(session: TransportSession)
9.     fun onFailure(errorMsg: String)
10. }
```

Используя **TransportSession**, который возвращается через **onSuccess(session: TransportSession)**, вы сможете в дальнейшем осуществлять связь (коммуникацию) с **сканер-кольцом WR15**.

Example

```
1. val sdk = M3Wr15Sdk.initialize(applicationContext, config)
2.
3. sdk.startS2PScan(object: DeviceFindResultListener {
4.     override fun onDeviceFound(device: BluetoothDevice) {
5.         sdk.stopS2PScan()
6.         sdk.connectToDevice(device, object:ConnectResultListener {
7.             override fun onSuccess(session: TransportSession) {
8.                 // Обработка успешного запроса
9.             }
10.            override fun onFailure(errorMsg: String) {
11.                // Обработка неудачного запроса
12.            }
13.        })
14.    }
15.
16.    override fun onScanFailed(errorCode: Int) {
17.        // Обработка ошибки сканирования
18.    }
19. })
```

6. Разрыв Соединения с WR15

6.1 Disconnect to WR15

Разрывает соединение с подключённым WR15.

```

1. /* TransportSession */
2.
3. fun disconnect(onSuccess: () -> Unit, onFailure: (errorMsg: String) -> Unit)
4. fun disconnect(listener: DisconnectResultListener)
```

6.2 Observe Disconnect

Отслеживает момент разрыва соединения.

```

1. /* TransportSession */
2.
3. fun addDisconnectListener(disconnectListener: DisconnectListener)
4. fun removeDisconnectListener(disconnectListener: DisconnectListener)
5.
6. fun interface DisconnectListener {
7.     fun onDisconnect()
8. }
```

Example

```

1. val disconnectListener = object: DisconnectListener {
2.     override fun onDisconnect() {
3.         // Обработка при отключении
4.     }
5. }
6. fun connectToDevice() {
7.     sdk.connectToDevice(device, object:ConnectResultListener {
8.         override fun onSuccess(session:TransportSession) {
9.             session.addDisconnectListener(disconnectListener)
10.        }
11.        override fun onFailure(errorMsg: String) {
12.            // Обработка неудачи
13.        }
14.    })
15. }
16.
17. fun disconnect() {
18.     if (sdk.isTransportSessionInitialized()) { // Попытка отключения только в случае, если сессия инициализирована
19.         val session = sdk.getTransportSession()
20.         session .disconnect(
21.             onSuccess = {
22.                 // Обработка успешного отключения
23.                 session.removeDisconnectListener(disconnectListener)
24.             },
25.             onFailure = { errorMsg ->
26.                 // Обработка неудачного отключения
27.             }
28.         )
29.     }
30. }
```

Чтобы предотвратить **утечку памяти** (memory leak), **обязательно** отменяйте регистрацию добавленных вами слушателей (**Listener**) с помощью метода **removeDisconnectListener()**, когда соединение разрывается или когда слушатель больше не используется

* В качестве параметра метода **removeDisconnectListener()** необходимо передать **тот же самый объект слушателя (Listener)**, который был зарегистрирован в **addDisconnectListener()**.

7. Прием Отсканированных Данных

7.1 Add Barcode Data

Для приёма декодированных данных штрих-кодов от WR15, необходимо зарегистрировать Listener.

```
1. /* TransportSession */
2.
3. fun addDecodingDataListener(decodingDataListener: DecodingDataListener)
4.
5. interface DecodingDataListener {
6.     fun onReceive(decodingData: ByteArray)
7. }
```

7.2 Remove Barcode Data

Отменяет регистрацию *BarcodeDataListener*.

```
1. /* TransportSession */
2.
3. fun removeBarcodeDataListener(barCodeDataListener: BarcodeDataListener)
```

Во избежание утечек памяти при разрыве соединения или когда Listener больше не используется, обязательно отменяйте его регистрацию.

* В параметр `removeBarcodeDataListener` необходимо передать тот же экземпляр Listener, который был зарегистрирован через `addBarcodeDataListener`.

8. Сведения об устройстве

Сведения о настройках устройства и сканера WR15 предоставляются в формате **data class DeviceState()**. За подробной информацией о полях и структуре DeviceState обратитесь к разделу [9. DeviceState](#).

8.1 Observe DeviceState

DeviceState предоставляется в формате Callback и повторно передаётся при каждом изменении настроек.

```
1. /* TransportSession */
2.
3. fun addDeviceStateListener(listener: DeviceStateListener)
4. fun removeDeviceStateListener(listener: DeviceStateListener)
5.
6. interface DeviceStateListener {
7.     fun onDeviceStateChanged(state: DeviceState)
8. }
9.
10. data class DeviceState(
11.     val deviceInfo: DeviceInfo = DeviceInfo(),
12.     val general: GeneralSettings = GeneralSettings(),
13.     val symbologies: Symbologies = Symbologies(),
14.     val basicFormat: BasicFormat = BasicFormat(),
15.     val readerSettings: ReaderSettings = ReaderSettings(),
16.     val devSettings: DevSettings = DevSettings()
17. )
```

8.2 Set Scanner Setting

Изменяет настройки сканера WR15.

For Kotlin

```
1. /* TransportSession */
2.
3. fun setSetting(setting: SettingCommand, onSuccess: () -> Unit, onFailure: (errorMsg: String) -> Unit)
```

For Java

```
1. /* TransportSession */
2.
3. fun setSetting(setting: SettingCommand, listener: SetSettingResultListener)
4.
5. interface SetSettingResultListener{
6.     fun onSuccess()
7.     fun onFailure(errorMsg: String)
8. }
```

8.2.1 SettingCommand

SettingCommand используется как параметр метода **setSetting()** и представляет объект, содержащий изменяемый параметр и соответствующие опции.

Создание **SettingCommand** выполняется с использованием объекта **Settings**.

8.2.2 Generate SettingCommand

Settings — это точка входа API, упрощающая создание **SettingCommand**.

```

1. object Settings {
2.     object Codabar {
3.         @JvmStatic
4.         fun setEnable(enable: Boolean): SettingCommand
5.         @JvmStatic
6.         fun setLength1(length: Int): SettingCommand
7.         ...
8.     }
9.
10.    object Code11 {
11.        @JvmStatic
12.        fun setEnable(enable: Boolean): SettingCommand
13.        @JvmStatic
14.        fun setLength1(length: Int): SettingCommand
15.        ...
16.    }
17.
18.    object General {
19.        @JvmStatic
20.        fun findMe(): SettingCommand
21.        @JvmStatic
22.        fun setAimer(enable: Boolean): SettingCommand
23.        @JvmStatic
24.        fun setIllumination(enable: Boolean): SettingCommand
25.        @JvmStatic
26.        fun setSound(type: SoundType): SettingCommand
27.        @JvmStatic
28.        fun setVibrate(enable: Boolean): SettingCommand
29.        ...
30.    }
31.    ... // Прочие настройки
32. }
```

За более подробным перечнем параметров обратитесь к разделу [10. Settings](#).

Example

For Kotlin

```

1. session.setSetting(
2.     Settings.EAN13.setEnable(true),
3.     onSuccess = {
4.         // Обработка успешного применения настроек
5.     },
6.     onFailure = { errorMsg ->
7.         // Обработка ошибки применения настроек
8.     }
9. )
```

For Java

```

1. session.setSetting(
2.     Settings.General.setSound(SoundType.OFF),
3.     object : SetSettingResultListener {
4.         override fun onFailure(errorMsg: String) {
5.             // Обработка ошибки применения настроек
6.         }
7.     }
8. )
```

```
7.  
8.     override fun onSuccess() {  
9.         // Обработка успешного применения настроек  
10.    }  
11.}  
12.)
```

8.2.3 Предостережение

Набор доступных параметров и значений опций отличается в зависимости от модели WR15 ring scanner (например, E4770, SE4107). Внимательно проверяйте документацию и запрашивайте только те настройки, которые поддерживаются соответствующей моделью.

Если для используемого WR15 ring scanner запрошена неподдерживаемая настройка, Callback метода *setSetting* обрабатывается через *onFailure()*.

9. DeviceState

DeviceState — объект, содержащий всю информацию о сканере. Для опций, не поддерживаемых используемым WR15 ring scanner, значения будут равны *null* или *Empty String*.

* Перечень типов Type доступен в разделе 10. Settings.

9.1 Strucutre

```
1. data class DeviceState(
2.     val deviceInfo: DeviceInfo = DeviceInfo(),
3.     val general: GeneralSettings = GeneralSettings(),
4.     val symbologies: Symbologies = Symbologies(),
5.     val basicFormat: BasicFormat = BasicFormat(),
6.     val readerSettings: ReaderSettings = ReaderSettings(),
7.     val devSettings: DevSettings = DevSettings()
8. )
```

9.1.1 DeviceInfo

```
1. data class DeviceInfo(
2.     val name: String = "",
3.     val macAddress: String = "",
4.     val model: String = "",
5.     val serialNumber: String = "",
6.     val fwVersion: String = "",
7.     val releaseDate: String = "",
8.     val batteryPercent: Int? = null,
9.     val scannerId: ScannerIdType? = null,
10.    val btMode: BtModeType? = null,
11. )
```

9.1.2 GeneralSettings

```
1. data class GeneralSettings(
2.     val sound: SoundType? = null,
3.     val vibrateEnabled: Boolean? = null,
4.     val ledEnabled: Boolean? = null,
5.     val buttonEnabled: Boolean? = null,
6.     val aimEnabled: Boolean? = null,
7.     val illuminationEnabled: Boolean? = null,
8. )
```

9.1.3 Symbologies

```
1. data class Symbologies(
2.     val australianPostalVals: AustralianVals = AustralianVals(),
3.     val aztecVals: AztecVals = AztecVals(),
4.     val chinese2Of5Vals: Chinese2Of5Vals = Chinese2Of5Vals(),
5.     val codabarVals: CodabarVals = CodabarVals(),
6.     val code11Vals: Code11Vals = Code11Vals(),
7.     val code39Vals: Code39Vals = Code39Vals(),
8.     val code93Vals: Code93Vals = Code93Vals(),
9.     val code128Vals: Code128Vals = Code128Vals(),
10.    val compositeVals: CompositeVals = CompositeVals(),
11.    val dataMatrixVals: DataMatrixVals = DataMatrixVals(),
12.    val discrete2Of5Vals: Discrete2Of5Vals = Discrete2Of5Vals(),
13.    val dotCodeVals: DotCodeVals = DotCodeVals(),
```

```

14. val ean8Vals: Ean8Vals = Ean8Vals(),
15. val ean13Vals: Ean13Vals = Ean13Vals(),
16. val gs1DataBar14Vals: Gs1DataBar14Vals = Gs1DataBar14Vals(),
17. val gs1DataBarExpandedVals: Gs1DataBarExpandedVals = Gs1DataBarExpandedVals(),
18. val gs1DataBarLimitedVals: Gs1DataBarLimitedVals = Gs1DataBarLimitedVals(),
19. val gs1128Vals: Gs1128Vals = Gs1128Vals(),
20. val hanXinVals: HanXinVals = HanXinVals(),
21. val interleaved2Of5Vals: Interleaved2Of5Vals = Interleaved2Of5Vals(),
22. val isbt128Vals: Isbt128Vals = Isbt128Vals(),
23. val japanesePostalVals: JapanesePostalVals = JapanesePostalVals(),
24. val korean3Of5Vals: Korean3Of5Vals = Korean3Of5Vals(),
25. val matrix2Of5Vals: Matrix2Of5Vals = Matrix2Of5Vals(),
26. val maxiCodeVals: MaxiCodeVals = MaxiCodeVals(),
27. val microPdf417Vals: MicroPdf417Vals = MicroPdf417Vals(),
28. val microQrCodeVals: MicroQrCodeVals = MicroQrCodeVals(),
29. val msiVals: MsiVals = MsiVals(),
30. val netherlandsKixVals: NetherlandsKixVals = NetherlandsKixVals(),
31. val pdf417Vals: Pdf417Vals = Pdf417Vals(),
32. val qrCodeVals: QrCodeVals = QrCodeVals(),
33. val ukPostalVals: UkPostalVals = UkPostalVals(),
34. val upcAVals: UpcAVals = UpcAVals(),
35. val upcE1Vals: UpcE1Vals = UpcE1Vals(),
36. val upcEanVals: UpcEanVals = UpcEanVals(),
37. val upcEVals: UpcEVals = UpcEVals(),
38. val upuFicsPostalVals: UpuFicsPostalVals = UpuFicsPostalVals(),
39. val usPlanetVals: UsPlanetVals = UsPlanetVals(),
40. val usPostnetVals: UsPostnetVals = UsPostnetVals(),
41. )

```

AustralianVals**Поддерживаемые сканеры : SE4107, SE5500**

```

1. data class AustralianVals(
2.     val enabled: Boolean? = null,
3. )

```

AztecVals**Поддерживаемые сканеры : SE4107, SE5500, E4770**

```

1. data class AztecVals(
2.     val enabled: Boolean? = null
3. )

```

Chinese2Of5Vals**Поддерживаемые сканеры : SE4107, SE5500**

```

1. data class Chinese2Of5Vals(
2.     val enabled: Boolean? = null
3. )

```

CodabarVals**Поддерживаемые сканеры : SE4107, SE5500, E4770**

```

1. data class CodabarVals(
2.     val enabled: Boolean? = null,
3.     val length1: Int? = null,
4.     val length2: Int? = null,
5.     val clsEditingStyle: Boolean? = null, //E4770 미지원
6.     val notisEditingStyle: Boolean? = null,
7.     val securityLevel: CodabarType.SecurityType? = null, //E4770 미지원
8. )

```

Code11ValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class Code11Vals(
2.     val enabled: Boolean? = null,
3.     val length1: Int? = null,
4.     val length2: Int? = null,
5.     val reportCheckDigit: Boolean? = null,
6.     val verifyCheckDigit: Code11Type.VerifyCheckDigitType? = null
7. )

```

Code39ValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class Code39Vals(
2.     val enabled: Boolean? = null,
3.     val triopticCode39: Boolean? = null, //E4770 — не поддерживается
4.     val length1: Int? = null,
5.     val length2: Int? = null,
6.     val reducedQuietZone: Boolean? = null, //E4770 — не поддерживается
7.     val convertToCode32: Boolean? = null,
8.     val fullAscii: Boolean? = null,
9.     val reportCheckDigit: Boolean? = null, // true = TRANSMIT
10.    val reportCode32Prefix: Boolean? = null,
11.    val securityLevel: Code39Type.SecurityType? = null, //E4770 — не поддерживается
12.    val verifyCheckDigit: Boolean? = null // true = ENABLE
13. )

```

Code93ValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class Code93Vals(
2.     val enabled: Boolean? = null,
3.     val length1: Int? = null,
4.     val length2: Int? = null
5. )

```

Code128ValsПоддерживающие сканеры: **SE4107, SE5500, E4770**

```

1. data class Code128Vals(
2.     val enabled: Boolean? = null,
3.     val length1: Int? = null,
4.     val length2: Int? = null,
5.     val reducedQuietZone: Boolean? = null, //E4770 — не поддерживается
6.     val ignoreFnc4: Boolean? = null, //E4770 — не поддерживается
7.     val checkIsbtTable: Boolean? = null, //E4770 — не поддерживается
8.     val isbt128ConcatMode: Code128Type.ISBT128ConcatModeType? = null, //E4770 — не поддерживается
9.     val securityLevel: Code128Type.SecurityType? = null, //E4770 — не поддерживается
10.    val emulationMode: Boolean? = null //E4770 — не поддерживается
11. )

```

CompositeValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class CompositeVals(
2.     val enabled: Boolean? = null, //SE4107, SE5500 — не поддерживается
3.     val compositeABEnabled: Boolean? = null, //E4770 — не поддерживается
4.     val compositeCEnabled: Boolean? = null, //E4770 — не поддерживается
5.     val tlc39Enable: Boolean? = null, //E4770 — не поддерживается
6.     val upcCompositeMode: CompositeType.UPCCCompositeModeType? = null //E4770 — не поддерживается

```

7.)

В модели E4770 единый параметр Enabled объединяет два параметра ABEnabled и CEnabled. В моделях SE4107 и SE5500 установка ABEnabled=true и CEnabled=true эквивалентна enabled=true

DataMatrixVals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class DataMatrixVals(  
2.     val enabled: Boolean? = null,  
3.     val inverse: DataMatrixType.InverseType? = null,  
4.     val decodeMirrorImages: DataMatrixType.DecodeMirrorImagesType? = null, //E4770 — не поддерживается  
5.     val gs1DataMatrix: Boolean? = null //E4770 — не поддерживается  
6. )
```

Discrete2Of5Vals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class Discrete2Of5Vals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null  
5. )
```

DotCodeVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class DotCodeVals(  
2.     val enabled: Boolean? = null,  
3.     val inverse: DotCodeType.InverseType? = null,  
4.     val mirror: DotCodeType.MirrorType? = null,  
5.     val prioritize: Boolean? = null  
6. )
```

Ean8Vals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class Ean8Vals(  
2.     val enabled: Boolean? = null  
3. )
```

Ean13Vals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class Ean13Vals(  
2.     val enabled: Boolean? = null  
3. )
```

Gs1DataBar14Vals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class Gs1DataBar14Vals(  
2.     val enabled: Boolean? = null  
3. )
```

Gs1DataBarExpandedVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class Gs1DataBarExpandedVals(  
2.     val enabled: Boolean? = null
```

3.)

Gs1DataBarLimitedVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class Gs1DataBarLimitedVals(  
2.     val enabled: Boolean? = null,  
3.     val securityLevel: Gs1DataBarLimitedType.SecurityType? = null  
4. )
```

Gs1128Vals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class Gs1128Vals(  
2.     val enabled: Boolean? = null  
3. )
```

HanXinVals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class HanXinVals(  
2.     val enabled: Boolean? = null,  
3.     val inverse: HanXinType.InverseType? = null  
4. )
```

Interleaved2Of5Vals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class Interleaved2Of5Vals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null,  
5.     val checkDigit: Interleaved2Of5Type.CheckDigitType? = null,  
6.     val reportCheckDigit: Boolean? = null,      // true = TRANSMIT  
7.     val securityLevel: Interleaved2Of5Type.SecurityType? = null, //E4770 — не поддерживается  
8.     val convertItf14ToEan13: Boolean? = null, //E4770 — не поддерживается  
9.     val reducedQuietZone: Boolean? = null //E4770 — не поддерживается  
10. )
```

Isbt128Vals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class Isbt128Vals(  
2.     val enabled: Boolean? = null  
3. )
```

JapanesePostalVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class JapanesePostalVals(  
2.     val enabled: Boolean? = null  
3. )
```

Korean3Of5Vals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class Korean3Of5Vals(  
2.     val enabled: Boolean? = null  
3. )
```

Matrix2Of5Vals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class Matrix2Of5Vals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null,  
5.     val redundancy: Boolean? = null, //E4770 — не поддерживается  
6.     val reportCheckDigit: Boolean? = null, // true = TRANSMIT  
7.     val verifyCheckDigit: Boolean? = null // true = ENABLE  
8. )
```

MaxiCode

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class MaxiCodeVals(  
2.     val enabled: Boolean? = null  
3. )
```

MicroPdf417Vals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class MicroPdf417Vals(  
2.     val enabled: Boolean? = null  
3. )
```

MicroQrCodeVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class MicroQrCodeVals(  
2.     val enabled: Boolean? = null  
3. )
```

MsiVals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class MsiVals(  
2.     val enabled: Boolean? = null,  
3.     val length1: Int? = null,  
4.     val length2: Int? = null,  
5.     val checkDigit: MsiType.CheckDigitType? = null,  
6.     val checkDigitScheme: MsiType.CheckDigitSchemeType? = null,  
7.     val reportCheckDigit: Boolean? = null // true = TRANSMIT  
8. )
```

NetherlandKixVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class NetherlandsKixVals(  
2.     val enabled: Boolean? = null  
3. )
```

Pdf417Vals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class Pdf417Vals(  
2.     val enabled: Boolean? = null  
3. )
```

QrCodeValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class QrCodeVals(
2.     val enabled: Boolean? = null
3. )

```

UkPostalValsПоддерживаемые сканеры : **SE4107, SE5500**

```

1. data class UkPostalVals(
2.     val enabled: Boolean? = null,
3.     val reportCheckDigit: Boolean? = null // true = TRANSMIT
4. )

```

UpcAValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class UpcAVals(
2.     val enabled: Boolean? = null,
3.     val reportCheckDigit: Boolean? = null,      // true = TRANSMIT
4.     val preamble: UpcAType.PreambleType? = null,
5.     val upcA2BitsSupplements: Boolean? = null,
6.     val upcA5BitsSupplements: Boolean? = null
7. )

```

UpcE1ValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class UpcE1Vals(
2.     val enabled: Boolean? = null,
3.     val convertToUpcA: Boolean? = null, // true = CONVERT //E4770 — не поддерживается
4.     val preamble: UpcE1Type.PreambleType? = null, //E4770 — не поддерживается
5.     val reportCheckDigit: Boolean? = null // true = TRANSMIT //E4770 — не поддерживается
6. )

```

UPpcEanValsПоддерживаемые сканеры : **SE4107, SE5500, E4770**

```

1. data class UpcEanVals(
2.     val reportEan8CheckDigit: Boolean? = null, // true = TRANSMIT
3.     val reportEan13CheckDigit: Boolean? = null, // true = TRANSMIT
4.
5.     val supplementalMode: UpcEanType.SupplementalModeType? = null, //E4770 — не поддерживается
6.     val supplementalAimIdFormat: UpcEanType.SupplementalAimIdFormatType? = null, //E4770 — не поддерживается
7.     val reducedQuietZone: Boolean? = null, //E4770 — не поддерживается
8.     val bookland: UpcEanType.BooklandType? = null, //E4770 — не поддерживается
9.     val uccCouponExtend: Boolean? = null, //E4770 — не поддерживается
10.    val couponReport: UpcEanType.CouponReportType? = null, //E4770 — не поддерживается
11.    val issnEan: Boolean? = null, //E4770 — не поддерживается
12.    val translateUpcAToEan13: Boolean? = null, //E4770 — не поддерживается
13.    val supplementalRedundancy: Int? = null, //E4770 — не поддерживается
14.    val supplemental2: Boolean? = null, //E4770 — не поддерживается
15.    val supplemental5: Boolean? = null, //E4770 — не поддерживается
16.
17.    val ean8ReadSupplements: Boolean? = null, //SE4107, SE5500 — не поддерживается
18.    val ean13ReadSupplements: Boolean? = null, // //SE4107, SE5500 — не поддерживается
19.    val ean8Supplemental2: Boolean? = null, // //SE4107, SE5500 — не поддерживается
20.    val ean13Supplemental2: Boolean? = null, // //SE4107, SE5500 — не поддерживается
21.    val ean8Supplemental5: Boolean? = null, // //SE4107, SE5500 — не поддерживается
22.    val ean13Supplemental5: Boolean? = null // //SE4107, SE5500 — не поддерживается

```

23.)

Функция Supplemental2, поддерживаемая в SE4107 и SE5500, в E4770 разделена на ean8Supplemental2 и ean8Supplemental5, а функция Supplemental5 — на ean13Supplemental2 и ean13Supplemental5.

Дополнительно: чтобы использовать параметры ean8Supplemental2 и ean8Supplemental5 в E4770, необходимо установить ean8ReadSupplements в значение true. Аналогично, для использования ean13Supplemental2 и ean13Supplemental5 следует установить ean13ReadSupplements в значение true.

UpcEVals

Поддерживаемые сканеры : **SE4107, SE5500, E4770**

```
1. data class UpcEVals(  
2.     val enabled: Boolean? = null,  
3.     val convertToUpcA: Boolean? = null,    // true = CONVERT  
4.     val preamble: UpcEType.PreambleType? = null,  
5.     val reportCheckDigit: Boolean? = null, // true = TRANSMIT  
6.     val upcE2BitsSupplements: Boolean? = null,  
7.     val upcE5BitsSupplements: Boolean? = null  
8. )
```

UpuFicsPostalVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class UpuFicsPostalVals(  
2.     val enabled: Boolean? = null  
3. )
```

UsPlanetVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class UsPlanetVals(  
2.     val enabled: Boolean? = null,  
3.     val reportCheckDigit: Boolean? = null // true = TRANSMIT  
4. )
```

UsPostnetVals

Поддерживаемые сканеры : **SE4107, SE5500**

```
1. data class UsPostnetVals(  
2.     val enabled: Boolean? = null  
3. )
```

9.1.4 BasicFormat

```
1. data class BasicFormat(  
2.     val transmitCodeId: BasicFormatType.TransmitCodeIdType? = null,  
3.     val endChar: BasicFormatType.EndCharType? = null,  
4.     val substringFormation: String = "",  
5.     val removeFnc: Boolean? = null,  
6.     val translateData: String = "",  
7.     val prefixPostfixAsAsciiHex: Boolean? = null,  
8.     val prefix: String = "",  
9.     val postfix: String = ""  
10. )
```

9.1.5 readerSettings

```
1. data class ReaderSettings(  
2.     val readMode: ReaderType.ReadModeType? = null,  
3.     val laserOnTime: Int? = null,  
4.     val inverse1D: ReaderType.Inverse1DType? = null,  
5.     val quietZoneLevelFor1D: ReaderType.QuietZoneLevelType? = null,  
6.     val poorQualityDecodeEffort: ReaderType.PoorQualityDecodeEffortType? = null,  
7. )
```

9.1.6 DevSettings

```
1. data class DevSettings(  
2.     val batteryCallback: Boolean? = null  
3. )
```

10. Settings

Settings — точка входа SDK для удобного создания **SettingCommand** по параметрам сканера. Опции по баркод-символикам и общие параметры ридера обёрнуты единообразными методами, что позволяет формировать безопасные команды простыми вызовами.

10.1 AustralianPostal

```
1. object AustralianPostal {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.2 Aztec

```
1. object Aztec {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.3 Chinese2of5

```
1. object Chinese2of5 {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.4 Codabar

```
1. object Codabar {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic  
6.     fun setLength1(length: Int): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setLength2(length: Int): SettingCommand  
10.  
11.    @JvmStatic //E4770 — не поддерживается  
12.    fun setCLSIEditing(enable: Boolean): SettingCommand  
13.  
14.    @JvmStatic  
15.    fun setNotisEditing(enable: Boolean): SettingCommand  
16.  
17.    @JvmStatic //E4770 — не поддерживается  
18.    fun setSecurityLevel(value: CodabarType.SecurityType): SettingCommand  
19. }
```

10.4.1 CodabarType

```
1. object CodabarType {  
2.     enum class SecurityType : TypeValue {  
3.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3  
4.     }
```

```
5. }
```

10.5 Code11

```
1. object Code11 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic  
6.     fun setLength1(length: Int): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setLength2(length: Int): SettingCommand  
10.  
11.    @JvmStatic  
12.    fun setReportCheckDigit(enable: Boolean): SettingCommand  
13.  
14.    @JvmStatic  
15.    fun setVerifyCheckDigit(value: Code11Type.VerifyCheckDigitType): SettingCommand  
16. }
```

10.5.1 Code11Type

```
1. object Code11Type {  
2.     enum class VerifyCheckDigitType : TypeValue {  
3.         DISABLE,  
4.         ONE_CHECK_DIGIT,  
5.         TWO_CHECK_DIGITS  
6.     }  
7. }
```

10.6 Code39

```
1. object Code39 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic //E4770 — не поддерживается  
6.     fun setTriopticCode39(enable: Boolean): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setLength1(length: Int): SettingCommand  
10.  
11.    @JvmStatic  
12.    fun setLength2(length: Int): SettingCommand  
13.  
14.    @JvmStatic //E4770 — не поддерживается  
15.    fun setReducedQuietZone(enable: Boolean): SettingCommand  
16.  
17.    @JvmStatic  
18.    fun setConvertToCode32(enable: Boolean): SettingCommand  
19.  
20.    @JvmStatic  
21.    fun setFullAscii(enable: Boolean): SettingCommand  
22.  
23.    @JvmStatic  
24.    fun setReportCheckDigit(enable: Boolean): SettingCommand
```

```
25.  
26. @JvmStatic  
27. fun setReportCode32Prefix(enable: Boolean): SettingCommand  
28.  
29. @JvmStatic //E4770 — не поддерживается  
30. fun setSecurityLevel(value: Code39Type.SecurityType): SettingCommand  
31.  
32. @JvmStatic  
33. fun setVerifyCheckDigit(enable: Boolean): SettingCommand  
34. }
```

10.6.1 Code39Type

```
1. object Code39Type {  
2.     enum class SecurityType : TypeValue {  
3.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3  
4.     }  
5. }
```

10.7 Code93

```
1. object Code93 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic  
6.     fun setLength1(length: Int): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setLength2(length: Int): SettingCommand  
10. }
```

10.8 Code128

```
1. object Code128 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic  
6.     fun setLength1(length: Int): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setLength2(length: Int): SettingCommand  
10.  
11.    @JvmStatic //E4770 — не поддерживается  
12.    fun setReducedQuietZone(enable: Boolean): SettingCommand  
13.  
14.    @JvmStatic //E4770 — не поддерживается  
15.    fun setIgnoreFnc4(enable: Boolean): SettingCommand  
16.  
17.    @JvmStatic //E4770 — не поддерживается  
18.    fun setCheckIsbtTable(enable: Boolean): SettingCommand  
19.  
20.    @JvmStatic //E4770 — не поддерживается  
21.    fun setIsbt128ConcatMode(value: Code128Type.ISBT128ConcatModeType): SettingCommand  
22.  
23.    @JvmStatic //E4770 — не поддерживается
```

```

24. fun setSecurityLevel(value: Code128Type.SecurityType): SettingCommand
25.
26. @JvmStatic //E4770 — не поддерживается
27. fun setEmulationMode(enable: Boolean): SettingCommand
28. }
```

10.8.1 Code128Type

```

1. object Code128Type {
2.     enum class SecurityType : TypeValue {
3.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
4.     }
5.
6.     enum class ISBT128ConcatModeType : TypeValue {
7.         ENABLE, DISABLE, AUTO
8.     }
9. }
```

10.9 Composite

```

1. object Composite {
2.     @JvmStatic //SE4107, SE5500 — не поддерживается
3.     fun setEnable(enable: Boolean): SettingCommand //SE4107, SE5500 — не поддерживается
4.
5.     @JvmStatic //E4770 — не поддерживается
6.     fun setABEnable(enable: Boolean): SettingCommand
7.
8.     @JvmStatic //E4770 — не поддерживается
9.     fun setCEnable(value: Boolean): SettingCommand
10.
11.    @JvmStatic //E4770 — не поддерживается
12.    fun setTlc39Enable(enable: Boolean): SettingCommand
13.
14.    @JvmStatic //E4770 — не поддерживается
15.    fun setUpcCompositeMode(value: CompositeType.UPCCompositeModeType): SettingCommand
16. }
```

- Настройка `setEnable(enable: Boolean)` одновременно применяет `setABEnable(enable)` и `setCEnable(enable)`. То есть для E4770 вызов `setEnable(true)` активирует АВ и С, а `setEnable(false)` деактивирует обе опции.

10.9.1 CompositeType

```

1. object CompositeType {
2.     enum class UPCCompositeModeType : TypeValue {
3.         NEVER_LINKED, ALWAYS_LINKED, AUTO_DISCRIMINATE
4.     }
5. }
```

10.10 DataMatrix

```

1. object DataMatrix {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setInverse(value: DataMatrixType.InverseType): SettingCommand
7. }
```

```

8.     @JvmStatic //E4770 — не поддерживается
9.     fun setDecodeMirrorImages(value: DataMatrixType.DecodeMirrorImagesType): SettingCommand
10.
11.    @JvmStatic //E4770 — не поддерживается
12.    fun setGs1DataMatrix(enable: Boolean): SettingCommand
13. }
```

10.10.1 DatatMatrixType

```

1. object DataMatrixType {
2.     enum class InverseType : TypeValue {
3.         REGULAR_ONLY, INVERSE_ONLY, AUTO
4.     }
5.
6.     enum class DecodeMirrorImagesType : TypeValue {
7.         NEVER, ALWAYS, AUTO
8.     }
9. }
```

10.11 Discrete2Of5

```

1. object Discrete2Of5 {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setLength1(length: Int): SettingCommand
7.
8.     @JvmStatic
9.     fun setLength2(length: Int): SettingCommand
10. }
```

10.12 DotCode

```

1. object DotCode {
2.     @JvmStatic //E4770 — не поддерживается
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic //E4770 — не поддерживается
6.     fun setInverse(value: DotCodeType.InverseType): SettingCommand
7.
8.     @JvmStatic //E4770 — не поддерживается
9.     fun setMirror(value: DotCodeType.MirrorType): SettingCommand
10.
11.    @JvmStatic //E4770 — не поддерживается
12.    fun setPrioritize(enable: Boolean): SettingCommand
13. }
```

10.12.1 DotCodeType

```

1. object DotCodeType {
2.     enum class InverseType : TypeValue { ENABLE, DISABLE, AUTODETECT }
3.     enum class MirrorType : TypeValue { REGULAR, INVERSE_ONLY, AUTODETECT }
4. }
```

10.13 EAN8

```
1. object EAN8 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.14 EAN13

```
1. object EAN13 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.15 Gs1DataBar14

```
1. object Gs1DataBar14 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.16 Gs1DataBarExpanded

```
1. object Gs1DataBarExpanded {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.17 Gs1DataBarLimited

```
1. object Gs1DataBarLimited {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic //E4770 — не поддерживается  
6.     fun setLimitedSecurityLevel(value: Gs1DataBarLimitedType.SecurityType): SettingCommand  
7. }
```

10.17.1 Gs1DataBarLimitedType

```
1. object Gs1DataBarLimitedType {  
2.     enum class SecurityType : TypeValue {  
3.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3  
4.     }  
5. }
```

10.18 Gs1128

```
1. object Gs1128 {  
2.     @JvmStatic
```

```

3.     fun setEnable(enable: Boolean): SettingCommand
4. }
```

10.19 HanXin

```

1. object HanXin {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setInverse(value: HanXinType.InverseType): SettingCommand
7. }
```

10.20 Interleaved2Of5

```

1. object Interleaved2Of5 {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setLength1(length: Int): SettingCommand
7.
8.     @JvmStatic
9.     fun setLength2(length: Int): SettingCommand
10.
11.    @JvmStatic
12.    fun setCheckDigit(value: Interleaved2Of5Type.CheckDigitType): SettingCommand
13.
14.    @JvmStatic
15.    fun setReportCheckDigit(enable: Boolean): SettingCommand
16.
17.    @JvmStatic //E4770 — не поддерживается
18.    fun setSecurityLevel(value: Interleaved2Of5Type.SecurityType): SettingCommand
19.
20.    @JvmStatic //E4770 — не поддерживается
21.    fun setConvertItf14ToEan13(enable: Boolean): SettingCommand
22.
23.    @JvmStatic //E4770 — не поддерживается
24.    fun setReducedQuietZone(enable: Boolean): SettingCommand
25. }
```

10.20.1 Interleaved2Of5Type

```

1. object Interleaved2Of5Type {
2.     enum class CheckDigitType : TypeValue {
3.         DISABLE,
4.         ENABLE,
5.         USS_CHECK_DIGIT, //E4770 — не поддерживается
6.         OPCC_CHECK_DIGIT //E4770 — не поддерживается
7.     }
8.
9.     enum class SecurityType : TypeValue {
10.         LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
11.     }
12. }
```

10.21 ISBT128

```
1. object ISBT128 {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.22 JapanesePostal

```
1. object JapanesePostal {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.23 Korean3of5

```
1. object Korean3of5 {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.24 Matrix2of5

```
1. object Matrix2Of5 {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic  
6.     fun setLength1(length: Int): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setLength2(length: Int): SettingCommand  
10.  
11.    @JvmStatic //E4770 — не поддерживается  
12.    fun setRedundancy(enable: Boolean): SettingCommand  
13.  
14.    @JvmStatic  
15.    fun setReportCheckDigit(enable: Boolean): SettingCommand  
16.  
17.    @JvmStatic  
18.    fun setVerifyCheckDigit(enable: Boolean): SettingCommand  
19. }
```

10.25 MaxiCode

```
1. object MaxiCode {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.26 MicroPdf417

```
1. object MicroPdf417 {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand
```

```
4. }
```

10.27 MicroQrCode

```
1. object MicroQrCode {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.28 MSI

```
1. object MSI {  
2.     @JvmStatic  
3.     fun setEnable(enable: Boolean): SettingCommand  
4.  
5.     @JvmStatic  
6.     fun setLength1(length: Int): SettingCommand  
7.  
8.     @JvmStatic  
9.     fun setLength2(length: Int): SettingCommand  
10.  
11.    @JvmStatic  
12.    fun setCheckDigit(value: MsiType.CheckDigitType): SettingCommand  
13.  
14.    @JvmStatic  
15.    fun setCheckDigitScheme(value: MsiType.CheckDigitSchemeType): SettingCommand  
16.  
17.    @JvmStatic  
18.    fun setReportCheckDigit(enable: Boolean): SettingCommand  
19. }
```

10.28.1 MsiType

```
1. object MsiType {  
2.     enum class CheckDigitType : TypeValue {  
3.         ONE_CHECK_DIGIT, TWO_CHECK_DIGITS  
4.     }  
5.  
6.     enum class CheckDigitSchemeType : TypeValue {  
7.         MOD_10_10, MOD_10_11  
8.     }  
9. }
```

10.29 NetherlandsKix

```
1. object NetherlandsKix {  
2.     @JvmStatic //E4770 — не поддерживается  
3.     fun setEnable(enable: Boolean): SettingCommand  
4. }
```

10.30 Pdf417

```
1. object Pdf417 {  
2.     @JvmStatic //E4770 — не поддерживается
```

```

3.     fun setEnable(enable: Boolean): SettingCommand
4. }
```

10.31 QrCode

```

1. object QrCode {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4. }
```

10.32 UKPostal

```

1. object UKPostal {
2.     @JvmStatic //E4770 — не поддерживается
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic //E4770 — не поддерживается
6.     fun setReportCheckDigit(enable: Boolean): SettingCommand
7. }
```

10.33 UPCA

```

1. object UPCA {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setReportCheckDigit(enable: Boolean): SettingCommand
7.
8.     @JvmStatic
9.     fun setPreamble(value: UpcAType.PreambleType): SettingCommand
10.
11.    @JvmStatic
12.    fun setSupplemental2Bit(enable: Boolean): SettingCommand
13.
14.    @JvmStatic
15.    fun setSupplemental5Bit(enable: Boolean): SettingCommand
16. }
```

10.33.1 UpcAType

```

1. object UpcAType {
2.     enum class PreambleType : TypeValue {
3.         NONE, SYS_CHAR, COUNTRY_SYS_CHAR
4.     }
5. }
```

10.34 UPCE1

```

1. object UPCE1 {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic //E4770 — не поддерживается
```

```

6. fun setConvertToUpcA(enable: Boolean): SettingCommand
7.
8. @JvmStatic //E4770 — не поддерживается
9. fun setPreamble(value: UpcE1Type.PreambleType): SettingCommand
10.
11. @JvmStatic //E4770 — не поддерживается
12. fun setReportCheckDigit(enable: Boolean): SettingCommand
13. }
```

10.34.1 UpcE1Type

```

1. object UpcE1Type {
2.     enum class PreambleType : TypeValue {
3.         NONE, SYS_CHAR, COUNTRY_SYS_CHAR
4.     }
5. }
```

10.35 UPCEan

```

1. object UPCEan {
2.     @JvmStatic
3.     fun setReportEan8CheckDigit(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setReportEan13CheckDigit(enable: Boolean): SettingCommand
7.
8.     @JvmStatic //E4770 — не поддерживается
9.     fun setSupplementalMode(value: UpcEanType.SupplementalModeType): SettingCommand
10.
11.    @JvmStatic //E4770 — не поддерживается
12.    fun setSupplementalAimIdFormat(value: UpcEanType.SupplementalAimIdFormatType): SettingCommand
13.
14.    @JvmStatic //E4770 — не поддерживается
15.    fun setReducedQuietZone(enable: Boolean): SettingCommand
16.
17.    @JvmStatic //E4770 — не поддерживается
18.    fun setBookland(value: UpcEanType.BooklandType): SettingCommand
19.
20.    @JvmStatic //E4770 — не поддерживается
21.    fun setUccCouponExtend(enable: Boolean): SettingCommand
22.
23.    @JvmStatic //E4770 — не поддерживается
24.    fun setCouponReport(value: UpcEanType.CouponReportType): SettingCommand
25.
26.    @JvmStatic //E4770 — не поддерживается
27.    fun setIssnEan(enable: Boolean): SettingCommand
28.
29.    @JvmStatic //E4770 — не поддерживается
30.    fun setTranslateUpcAToEan13(enable: Boolean): SettingCommand
31.
32.    @JvmStatic //E4770 — не поддерживается
33.    fun setSupplementalRedundancy(value: Int): SettingCommand
34.
35.    @JvmStatic //E4770 — не поддерживается
36.    fun setEanSupplemental2(enable: Boolean): SettingCommand
37.
38.    @JvmStatic //E4770 — не поддерживается
39.    fun setEanSupplemental5(enable: Boolean): SettingCommand
40.
```

```

41. @JvmStatic //SE4107, SE5500 — не поддерживается
42. fun setEan8ReadSupplements(enable: Boolean): SettingCommand
43.
44. @JvmStatic //SE4107, SE5500 — не поддерживается
45. fun setEan13ReadSupplements(enable: Boolean): SettingCommand
46.
47. @JvmStatic //SE4107, SE5500 — не поддерживается
48. fun setEan8Supplemental2(enable: Boolean): SettingCommand
49.
50. @JvmStatic //SE4107, SE5500 — не поддерживается
51. fun setEan13Supplemental2(enable: Boolean): SettingCommand
52.
53. @JvmStatic //SE4107, SE5500 — не поддерживается
54. fun setEan8Supplemental5(enable: Boolean): SettingCommand
55.
56. @JvmStatic //SE4107, SE5500 — не поддерживается
57. fun setEan13Supplemental5(enable: Boolean): SettingCommand
58. }

```

- Методы `setEan8Supplemental2(enable)` и `setEan8Supplemental5(enable)` работают только при предварительной установке `setEan8ReadSupplements(true)`.
- Методы `setEan13Supplemental2(enable)` и `setEan13Supplemental5(enable)` работают только при предварительной установке `setEan13ReadSupplements(true)`.

10.35.1 UpcEanType

```

1. object UpcEanType {
2.     enum class SupplementalModeType : TypeValue {
3.         IGNORE_SUPPLEMENTS,
4.         DECODE_WITH_SUPPLEMENTS,
5.         AUTODISCRIMINATE_SUPPLEMENTS,
6.         SMART_SUPPLEMENTAL_MODE,
7.         ENABLE_378_379,
8.         ENABLE_978_979,
9.         ENABLE_414_419_434_439,
10.        ENABLE_977,
11.        ENABLE_491,
12.        USER_PROGRAMMABLE_TYPE1,
13.        USER_PROGRAMMABLE_TYPE1_2,
14.        SMART_PLUS_USER_PROGRAMMABLE1,
15.        SMART_PLUS_USER_PROGRAMMABLE1_2
16.    }
17.
18.    enum class SupplementalAimIdFormatType : TypeValue {
19.        SEPARATE, COMBINED, SEPARATE_TRANSMISSION
20.    }
21.
22.    enum class CouponReportType : TypeValue {
23.        OLD_FORMAT, NEW_FORMAT, AUTODISCRIMINATE
24.    }
25.
26.    enum class BooklandType : TypeValue {
27.        ISBN10, ISBN13
28.    }
29. }

```

10.36 UPCE

```

1. object UPCE {
2.     @JvmStatic
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic
6.     fun setConvertToUpcA(enable: Boolean): SettingCommand
7.
8.     @JvmStatic
9.     fun setPreamble(value: UpcEType.PreambleType): SettingCommand
10.
11.    @JvmStatic
12.    fun setReportCheckDigit(enable: Boolean): SettingCommand
13.
14.    @JvmStatic
15.    fun setSupplemental2Bit(enable: Boolean): SettingCommand
16.
17.    @JvmStatic
18.    fun setSupplemental5Bit(enable: Boolean): SettingCommand
19. }
```

10.36.1 UpcEType

```

1. object UpcEType {
2.     enum class PreambleType : TypeValue {
3.         NONE, SYS_CHAR, COUNTRY_SYS_CHAR
4.     }
5. }
```

10.37 UpuFicsPostal

```

1. object UpuFicsPostal {
2.     @JvmStatic //E4770 — не поддерживается
3.     fun setEnable(enable: Boolean): SettingCommand
4. }
```

10.38 USPlanet

```

1. object USPlanet {
2.     @JvmStatic //E4770 — не поддерживается
3.     fun setEnable(enable: Boolean): SettingCommand
4.
5.     @JvmStatic //E4770 — не поддерживается
6.     fun setReportCheckDigit(enable: Boolean): SettingCommand
7. }
```

10.39 USPostnet

```

1. object USPostnet {
2.     @JvmStatic //E4770 — не поддерживается
3.     fun setEnable(enable: Boolean): SettingCommand
4. }
```

10.40 General

```

1. object General {
2.     @JvmStatic
3.     fun findMe(): SettingCommand
4.
5.     @JvmStatic
6.     fun setAimer(enable: Boolean): SettingCommand
7.
8.     @JvmStatic
9.     fun setIllumination(enable: Boolean): SettingCommand
10.
11.    @JvmStatic
12.    fun setSound(type: SoundType): SettingCommand
13.
14.    @JvmStatic
15.    fun setVibrate(enable: Boolean): SettingCommand
16.
17.    @JvmStatic
18.    fun setLedEnable(enable: Boolean): SettingCommand
19.
20.    @JvmStatic
21.    fun setButtonEnable(enable: Boolean): SettingCommand
22.
23.    @JvmStatic
24.    fun setWithCodeType(enable: Boolean): SettingCommand
25. }
```

10.40.1 SoundType

```

1. enum class SoundType : TypeValue {
2.     ON, OFF, MUTE_ON_FAILED_SCAN
3. }
```

10.41 ReaderParams

```

object ReaderParams {
    @JvmStatic
    fun setReadMode(value: ReaderType.ReadModeType): SettingCommand

    /** value is in deciseconds (seconds × 10). e.g., 1s -> 10, 1.5s -> 15, 10s -> 100. */
    @JvmStatic
    fun setLaserOnTime(value: Int): SettingCommand

    @JvmStatic
    fun setInverse1D(value: ReaderType.Inverse1DType): SettingCommand

    @JvmStatic
    fun setQuietZoneLevelFor1D(value: ReaderType.QuietZoneLevelType): SettingCommand

    @JvmStatic
    fun setPoorQualityDecodeEffort(value: ReaderType.PoorQualityDecodeEffortType): SettingCommand
}
```

10.41.1 ReaderType

```

1. object ReaderType {
2.     enum class Inverse1DType : TypeValue {
3.         REGULAR_ONLY, INVERSE_ONLY, INVERSE_AUTODETECT
4.     }
5. }
```

```

6. enum class QuietZoneLevelType : TypeValue {
7.     LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
8. }
9.
10. enum class PoorQualityDecodeEffortType : TypeValue {
11.     LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3
12. }
13.
14. enum class ReadModeType : TypeValue {
15.     ASYNC, SYNC
16. }
17. }
```

10.42 BasicDataFormat

```

1. object BasicDataFormat {
2.     @JvmStatic
3.     fun setTransmitCodeID(value: BasicFormatType.TransmitCodeIdType): SettingCommand
4.
5.     @JvmStatic
6.     fun setEndChar(value: BasicFormatType.EndCharType): SettingCommand
7.
8.     @JvmStatic
9.     fun setSubString(value: String): SettingCommand
10.
11.    @JvmStatic
12.    fun setRemoveFnc(enable: Boolean): SettingCommand
13.
14.    @JvmStatic
15.    fun setTranslateData(value: String): SettingCommand
16.
17.    @JvmStatic
18.    fun setPrefixPostfixAsAsciiHex(enable: Boolean): SettingCommand
19.
20.    @JvmStatic
21.    fun setPrefix(value: String): SettingCommand
22.
23.    @JvmStatic
24.    fun setPostfix(value: String): SettingCommand
25. }
```

10.42.1 BasicFormatType

```

1. object BasicFormatType {
2.     enum class TransmitCodeIdType : TypeValue {
3.         NONE, AIM, SYMBOL
4.     }
5.
6.     enum class EndCharType : TypeValue {
7.         ENTER, TAB, NONE
8.     }
9. }
```

10.42.2 String Value Format

Все значения String должны задаваться в соответствии с форматом ниже; специальные символы также должны быть включены.

- *setSubstring(String)*

Данные, выводимые сканером, поступают обрезанными в соответствии с настроенными индексами.

Формат ввода: начальный индекс,конечный индекс.

ex) 3,5

- *setTranslateData(String)*

Преобразует байты, выводимые сканером, сопоставляя исходный байт с заменяющим байтом.

Формат ввода: [исходное значение,заменяющее значение];[исходное значение,заменяющее значение] ... максимум до 8 пар.

ex) 5B,28;5D,29

- *setPrefix(String)*

Настроенные байты добавляются в начало выходных данных сканера как префикс.

Формат ввода: до 16 символов (16 bytes).

ex) abcdef

- *setPostfix(String)*

Настроенные байты добавляются в конец выходных данных сканера как суффикс.

Формат ввода: до 16 символов (16 bytes).

ex) abcdef

10.43 Dev

```
1. object Dev {  
2.     @JvmStatic  
3.     fun setBatteryCallback(enable: Boolean): SettingCommand  
4. }
```

11. M3Utils

Это набор полезных функций при разработке приложения с использованием WR15 SDK.

1. toUtf8String

```
1. /* M3Utils */  
2.  
3. @JvmOverloads  
4. @JvmStatic  
5. fun toUtf8String(bytes: ByteArray, charset: Charset =Charsets.UTF_8): String
```

Декодирует массив байтов в String с использованием указанного набора символов; по умолчанию — UTF-8.

2. isDecimalPair

```
1. /* M3Utils */  
2.  
3. @JvmStatic  
4. fun isDecimalPair(input: String): Boolean
```

Проверяет, соответствует ли ввод String формату “Int,Int”.

Может использоваться для валидации параметра при вызове Settings.BasicDataFormat.setSubString(String).

3. sHexPairsSequenceAndMaxLength8

```
1. /* M3Utils */  
2.  
3. @JvmStatic  
4. fun isHexPairsSequenceAndMaxLength8(value: String): Boolean
```

Проверяет, что ввод соответствует формату “hh, hh; hh, hh” и содержит не более 8 пар, разделённых символом ‘;’.

Может использоваться для валидации параметра при вызове Setting.BasicDataFormat.setTranslateData(String).

4. generateQrBitmap(DeviceState, Int)

```
1. /* M3Utils */  
2.  
3. @JvmOverloads  
4. @JvmStatic  
5. fun generateQrBitmap(deviceState: DeviceState, size: Int = 512): Bitmap?
```

Создаёт QR-код настроек WR15 на основе **DeviceState**.

- deviceState: **DeviceState** — объект для генерации в Qr
- size: Int — размер Qr-кода (по умолчанию 512)