



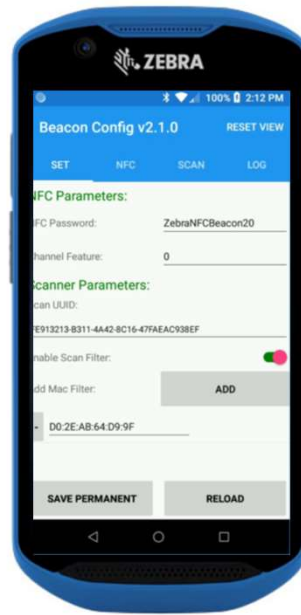
Zebra NFC Toolbox Service and Sample User Android Applications User Manual

October 2023



Custom Beacon Configuration

BLE Toolbox Application (using NFC)



Physical NFC tap to connect,
Beacon LEDs & App feedback

- Install application on Android device
- Device must have NFC enabled
- Enable location services
- Run the configurator application
- Select “NFC” tab and operation
 - Read or configure parameter
 - Read or configure password
 - Broadcast or sleep
 - Clear screen data; stop operation
- Tap top of BLE beacon to interact
- Scan is a BLE scanner
- Log shows command execution

SB1100 Enhanced Asset Beacon Product Overview

Small, BLE 5.x, Waterproof, NFC

SB1100



- Small, coin cell beacons ideal for asset tracking in all forms
- BLE 5.x and **NFC** in a Waterproof housing

- Optimized for long battery life within industrial environments
- Cleanable with chemicals common in healthcare facilities

- Low RF transmit powers for increased location accuracy and improved performance in high-density deployments

SB1100 Enhanced Asset Beacon Specifications



SIZE

1.46 in. x 1.06 in. x 0.47 in. (Incl. VHB Tape)



TRANSMIT POWER

-7 dBm to -30 dBm (EIRP)

Configurable



WEIGHT

8 g (0.28 oz)



ANTENNA TYPE

Omni directional



BATTERY

Fixed CR2032 220 mAh



OPTIONAL ACCESSORIES

Slotted mounting plate (for non-flat surfaces)

Wrist mounting plate (to be wearable)



- Sold only as kit of 25 beacons
- IP67 healthcare cleanable, 0-40°C
- BLE 5.x with Tx channel select
- **13.56 MHz NFC config. via Android App**

- Beacon attaches to flat surface with two-sided, closed-cell 3M vHB tape which comes pre-attached to the beacon
- Beacon attaches to flat surfaces and other surfaces using optional mounting plates

- 2-year battery life at 2s interval (5yr @ 5 sec)
- Non-replaceable lithium battery
- Reduced EMI design-can be attached directly to sensitive medical devices

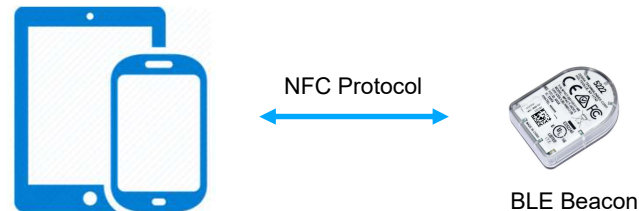
Zebra IoT BLE Beacon Configuration via NFC

The BLE beacon configuration process

Question: How do you configure the Zebra BLE beacon via NFC?

Answer 1: You configure the device with the standalone Zebra NFC Toolbox Android app.

- The standalone Toolbox App uses NFC to configure beacons
- The standalone Toolbox App requires manual settings and actions
- The standalone Toolbox App can't be integrated with existing customer solution automatically



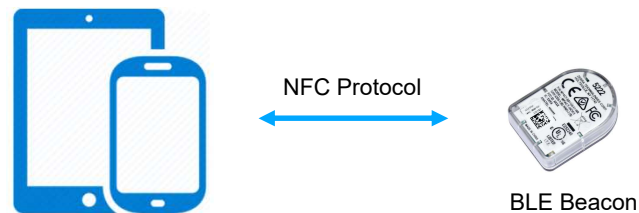
Zebra IoT BLE Beacon Configuration via NFC

The BLE beacon configuration process

Question: How do you configure the Zebra BLE beacon via NFC?

Answer 2: You can configure the device with your own app via the NFC Toolbox Service Android App.


- A User App is required to send the configuration request to the NFC Toolbox Service App
- The NFC Toolbox Service App uses NFC to configure beacons
- The NFC Toolbox Service App has no UI
- The NFC Toolbox Service App sends the configuration results to the User App
- The NFC Toolbox Service App can be integrated with the existing customer solution



The NFC Toolbox Service App Overview


The NFC Toolbox Service Android Mobile App Capabilities

The App Functionality

- A light-weight Android app to configure Zebra BLE Beacon MPACT-SB1100-01-WR via NFC
 - Android intent is used between the User App and the NFC Toolbox Service App
 - The NFC Toolbox Service App is a hidden app without any UI
 - The NFC Toolbox Service App is a simple example that supports single or continuous configuration methods as an example or starting place for custom-built applications
 - Customers may incorporate and modify the app into their own solutions
 - A sample user app as well as source codes are provided to demonstrate how to use the NFC Toolbox Service App
- 


Zebra IoT BLE Beacon Configuration via NFC

The beacon configuration process

- User installs the NFC Toolbox Service App and the User App.
 - The beacon parameters can be changed manually or automatically inside the user app.
 - The user app puts together the configuration data into an intent and sends to the NFC Toolbox Service App.
 - The NFC Toolbox Service App acknowledges the user app via intent.
 - User taps the mobile or tablet to the BLE beacon to configure it.
 - The NFC Toolbox Service App transmits the configuration data to the beacon via NFC.
 - The NFC Toolbox Service App gets response from the beacon and sends the configuration result to the user app via intent.
 - The user app presents the result to user.
- 


NFC Toolbox Service App Functions

Functionality Provided by the app

- Main functions:
 - Activate the beacon to put it into broadcasting mode
 - Read the MAC address of the beacon
 - Activate and read MAC of the beacon
 - Put the beacon into sleep mode
 - Read parameters of the beacon
 - Change parameters of the beacon
 - Read NFC password of the beacon
 - Change NFC password of the beacon
 - Other functions:
 - Parameter range checking when changing beacon parameters
 - Single mode or continuous mode
 - Timeout in single mode
 - Stop current NFC process
- 

NFC Toolbox Service App Integration

How to communicate with the app

- The communication between the user app and the NFC Toolbox Service App is via intent broadcasting and receiving
 - Parameters are embedded in the intent for data exchanging
 - Supported variables are defined in the file `CommonVariables.java` (In the sample codes)
 - Only 2 variables need to be changed to make the user app work with the NFC Toolbox Service App
 - `CUSTOMER_APP_INTENT_ACTIVITY_ID`
 - `CUSTOMER_APP_PKG_ID`
 - The code snippets are in the file `MainActivity.java`, function `createIntent()`
 - User app's `AndroidManifest.xml` also needs to be updated/changed
- 

NFC Toolbox Service App Integration

What to change in the customer app for the integration

- AndroidManifest.xml:

- Get the package id (highlighted in the sample app)

- Change CUSTOMER_APP_PKG_ID in CommonVariables.java
- CUSTOMER_APP_PKG_ID = "com.zebra.nfc.userapp"

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  package="com.zebra.nfc.userapp">
```

- Add a “receiver” section, get the receiver activity id (highlighted in the sample app)

- Change CUSTOMER_APP_INTENT_ACTIVITY_ID in CommonVariables.java
- CUSTOMER_APP_INTENT_ACTIVITY_ID = "com.zebra.nfc.userapp.receiver"

```
<receiver
  android:name=".IntentReceiver"
  android:exported="true">
  <intent-filter android:priority="100">
    <action android:name="com.zebra.nfc.userapp.receiver" />
  </intent-filter>
</receiver>
```

NFC Toolbox Service App Integration

A simple Activate and Read MAC command integration

- In MainActivity.java:

- Use the action to create intent:

- CommonVariables.NFC_TOOLBOX_ACTION_ACTIVATE_BEACON_READ_MAC_ID

- Send out the intent via sendBroadcast():

- sendBroadcast(newIntent);

```
startNFCToolboxBroadcast(CommonVariables.NFC_TOOLBOX_ACTION_ACTIVATE_BEACON_READ_MAC_ID);
```

```
public void startNFCToolboxBroadcast(String action){  
    Intent newIntent = createIntent(action);  
    sendBroadcast(newIntent);  
}
```

```
public Intent createIntent(String action){  
    Parameters parameters = Parameters.GetParameters();  
    // Target intent  
    Intent serviceIntent = new Intent(CommonVariables.NFC_TOOLBOX_ACTIVITY_ID);  
    serviceIntent.setPackage(CommonVariables.NFC_TOOLBOX_PKGID);  
  
    // Intent compulsory parameters  
    serviceIntent.putExtra(CommonVariables.NFC_TOOLBOX_PARAM_ACTION, action);  
    serviceIntent.putExtra(CommonVariables.NFC_TOOLBOX_PARAM_ACTIVITY_ID, CommonVariables.CUSTOMER_APP_INTENT_ACTIVITY_ID);  
    serviceIntent.putExtra(CommonVariables.NFC_TOOLBOX_PARAM_PKGID, CommonVariables.CUSTOMER_APP_PKG_ID);  
  
    // Intent Continuous mode  
    serviceIntent.putExtra(CommonVariables.NFC_TOOLBOX_PARAM_CONTINUOUS_MODE, parameters.continuousMode);  
  
    // Intent timeout parameter  
    serviceIntent.putExtra(CommonVariables.NFC_TOOLBOX_PARAM_TIMEOUT, parameters.timeout);  
}
```

NFC Toolbox Service App Integration

A simple Activate and Read MAC command integration

- In IntentReceiver.java:
 - The NFC Toolbox Service App sends an intent to the user app to notify the result
 - The result intent is received in the function onReceive().

```
@Override
public void onReceive(Context context, Intent intent) {
    final String action = intent.getStringExtra(CommonVariables.NFC_TOOLBOX_PARAM_ACTION);
    final int resultCode = intent.getIntExtra(CommonVariables.NFC_TOOLBOX_ACTION_RESULT_CODE, CommonVariables.NFC_ACTION_FAILURE_NO_RESULT);
    final String resultMessage = intent.getStringExtra(CommonVariables.NFC_TOOLBOX_ACTION_RESULT_MESSAGE);
    final boolean continuousMode = intent.getBooleanExtra(CommonVariables.NFC_TOOLBOX_PARAM_CONTINUOUS_MODE, false);
    final String macid = intent.getStringExtra(CommonVariables.NFC_TOOLBOX_PARAM_MACID);

    PopupMessage(context, action, resultCode, macid);

    ShowResult(action, continuousMode, resultCode, resultMessage, macid);


    if(resultCode == CommonVariables.NFC_ACTION_FAILURE_SUCCESS)
    {
        if(action.equals(CommonVariables.NFC_TOOLBOX_ACTION_READ_BEACON_PARAMETERS))
            AppendBeaconParameters(intent);

        if(action.equals(CommonVariables.NFC_TOOLBOX_ACTION_READ_NFC_PASSWORD))
            AppendBeaconNfcPassword(intent);

        if(action.equals(CommonVariables.NFC_TOOLBOX_ACTION_CHANGE_NFC_PASSWORD))
            AppendBeaconNfcPasswordChange(intent);
    }
}
```

NFC Toolbox Service App Integration

Other commands integration

- Refer to the file MainActivity.java, function createIntent() for all the commands
 - Refer to file IntentReceiver.java for:
 - Error message interpretation
 - Result decoding
 - Parameter decoding
 - Refer to the file Parameters.java for the supported parameters
- 

NFC Toolbox Service App Integration

Parameter Valid Ranges

- Parameter values are checked with their valid ranges. If there is any error, an intent with invalid range error is sent to the user app:
 - `CommonVariables.NFC_ACTION_FAILURE_PARAMETER_INVALID_RANGE`

- Valid ranges as below:

`timeout` : 1000ms to 600000ms (1 second to 600 seconds)

`power` : valid values: 2,1,0,-3,-6,-9,-12,-15,-18,-21

`channel` : 1 to 7. 1: channel 39; 2: channel 38; 4: channel 37. 7: channel 37, 38, 39

`interval` : 100ms to 10000ms (0.1 second to 10 seconds)

`mode` : 2 or 3. 2: standard iBeacon mode; 3: MPact mode with battery info

`uuid` : standard 16-bytes UUID

`major` : 0 to 65535

`minor` : 0 to 65535

`channelFeature` : true or false

`continuousMode` : true or false

`nfcPassword` : 16-byte length String with character set: [a-zA-Z0-9]


`nfcPasswordToChange` : 16-byte length String with character set: [a-zA-Z0-9]

User NFC Sample App Overview



User NFC Sample App Overview

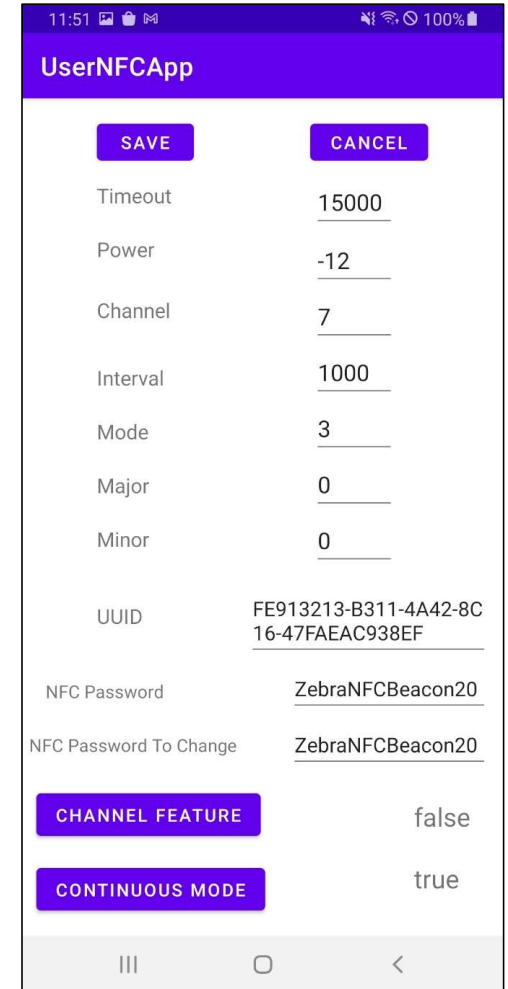
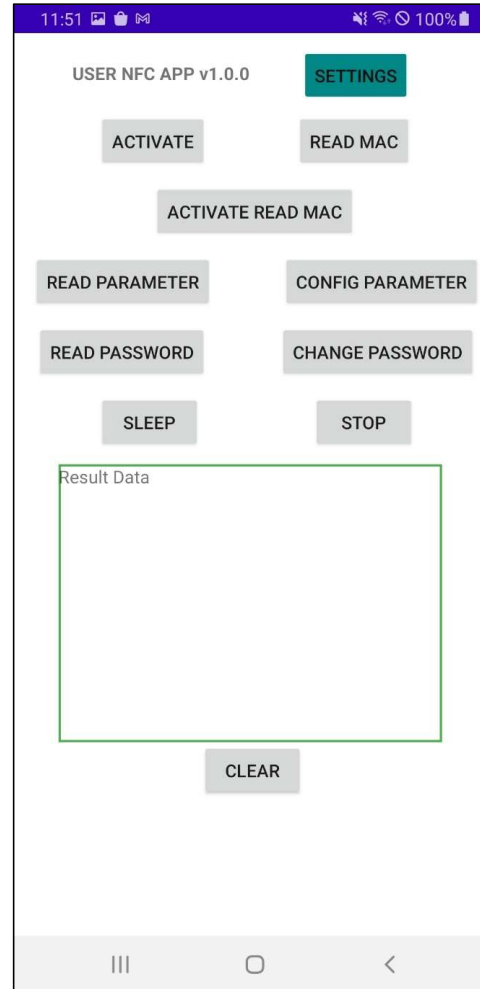
The App Functionality

- Provide a sample app as-is to demonstrate how to use the NFC Toolbox Service App
 - Not intended to be an end-user product but rather an example tool
 - Provide the sample source codes for development purpose
 - Has a simple settings page to change the parameters for demo purpose
- 

Sample User App Overview

The App is for demo purpose

- The main page
 - Each button presents a different function
 - In continuous mode, user can configure multiple NFC beacons in a run
 - The Clear button is to clear the contents inside the result box
- The settings page
 - User can change parameters
 - Continuous mode is enabled by default



Sample User App Function

Activate Beacon to broadcast mode

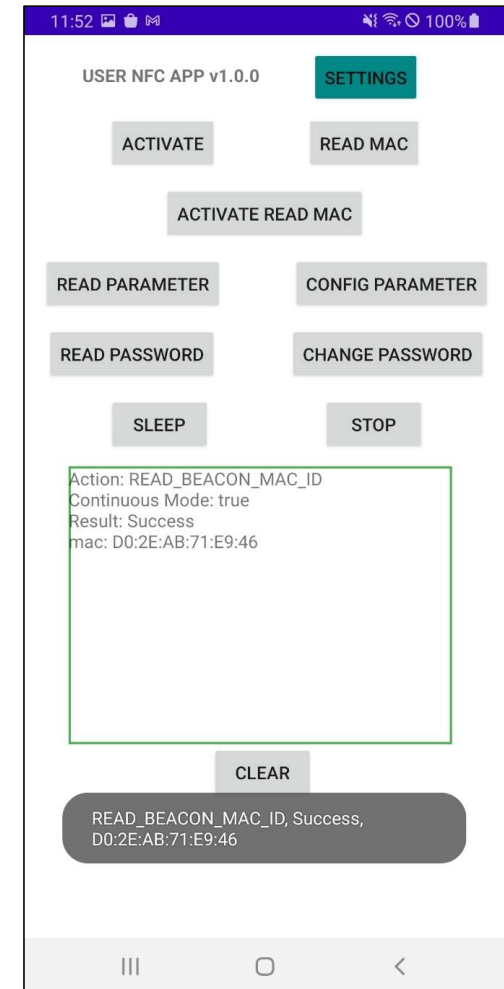
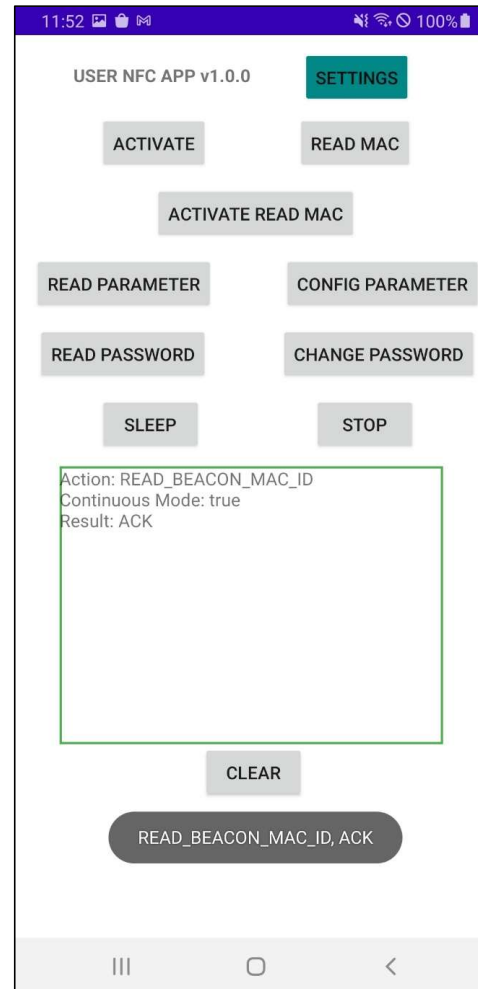
- The main page
 - Click the “Activate” button
- The result from the NFC Toolbox Service App



Sample User App Function

Read MAC of the Beacon

- The main page
 - Click the “Read MAC” button
- The result from the NFC Toolbox Service App



Sample User App Function

Activate and Read MAC

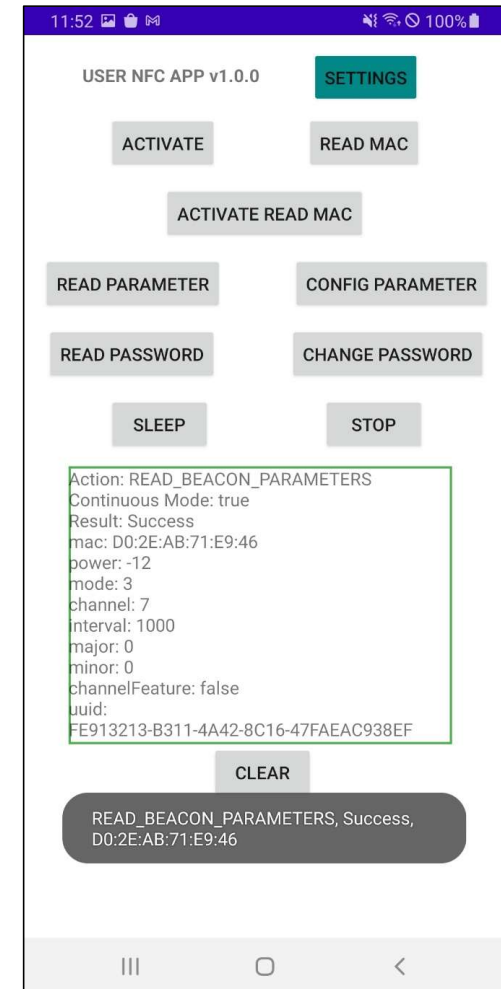
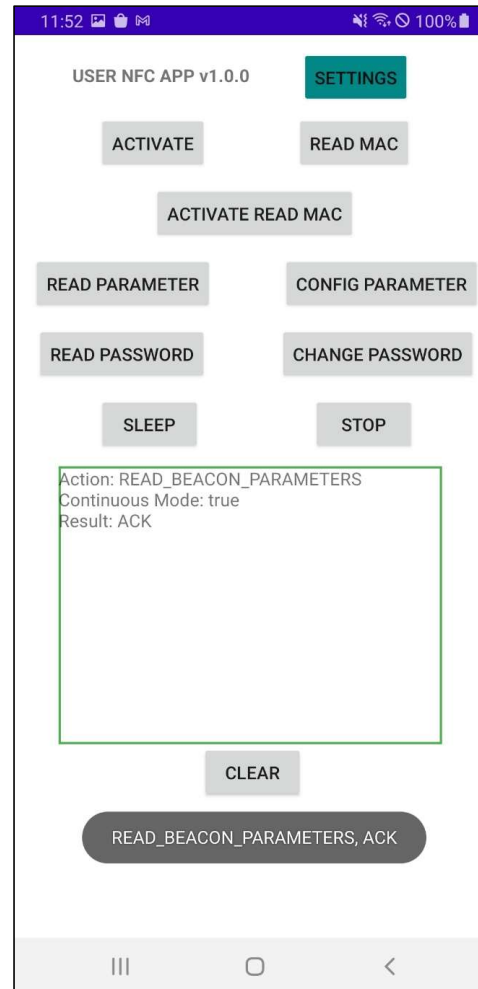
- The main page
 - Click the “Activate Read MAC” button
- The result from the NFC Toolbox Service App



Sample User App Function

Read parameters of the Beacon

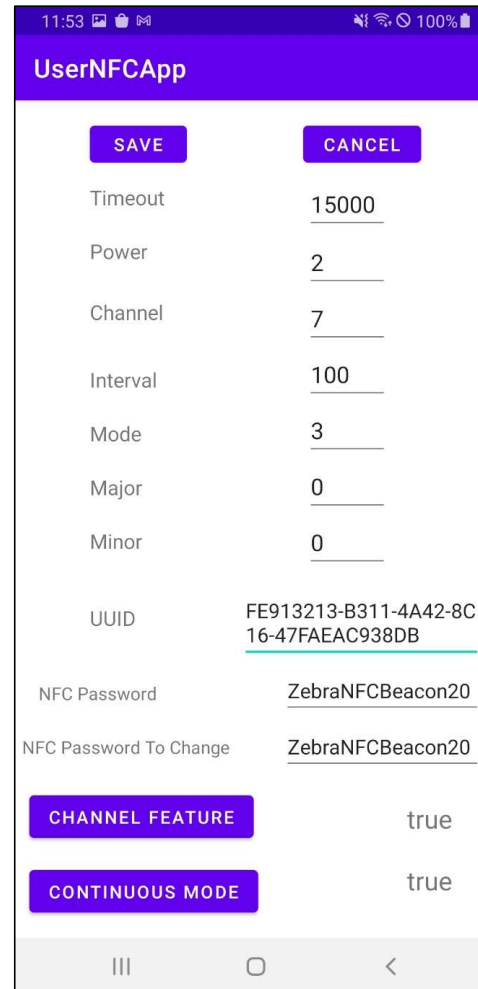
- The main page
 - Click the “Read Parameter” button
- The result from the NFC Toolbox Service App



Sample User App Function

Change Beacon parameters

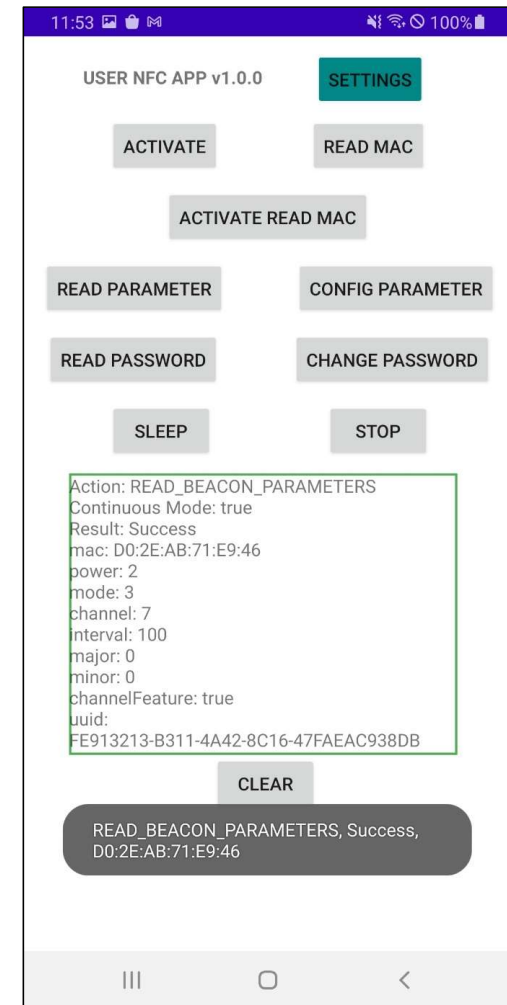
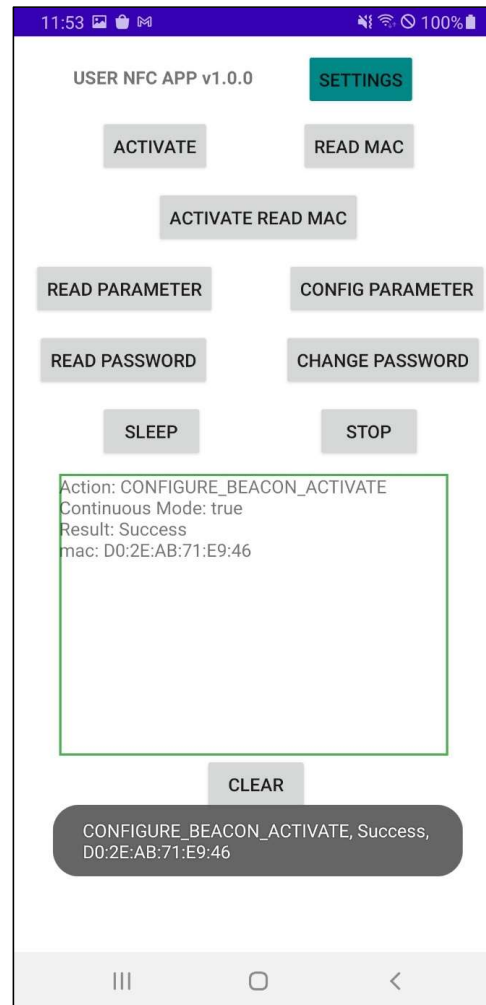
- The Settings page
 - Change the configuration parameters
- The main page
 - Click the “Config Parameter” button



Sample User App Function

Change Beacon parameters

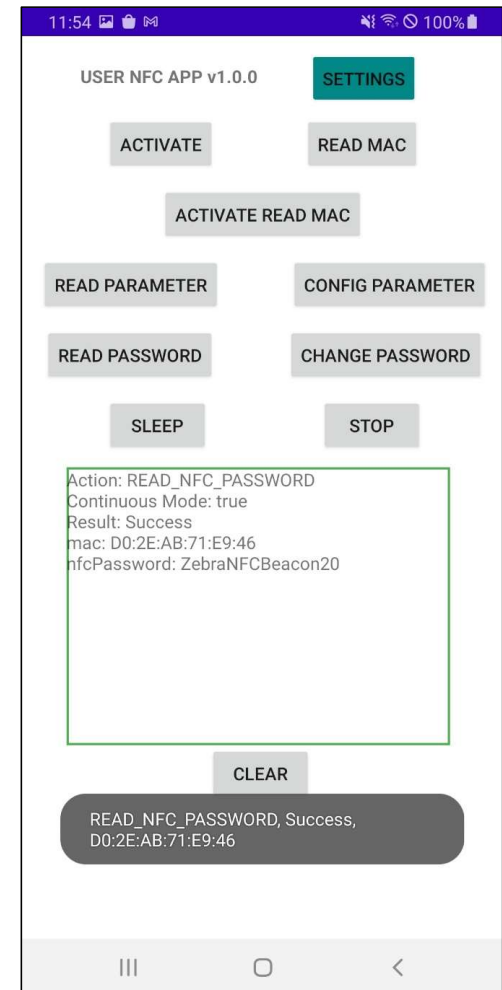
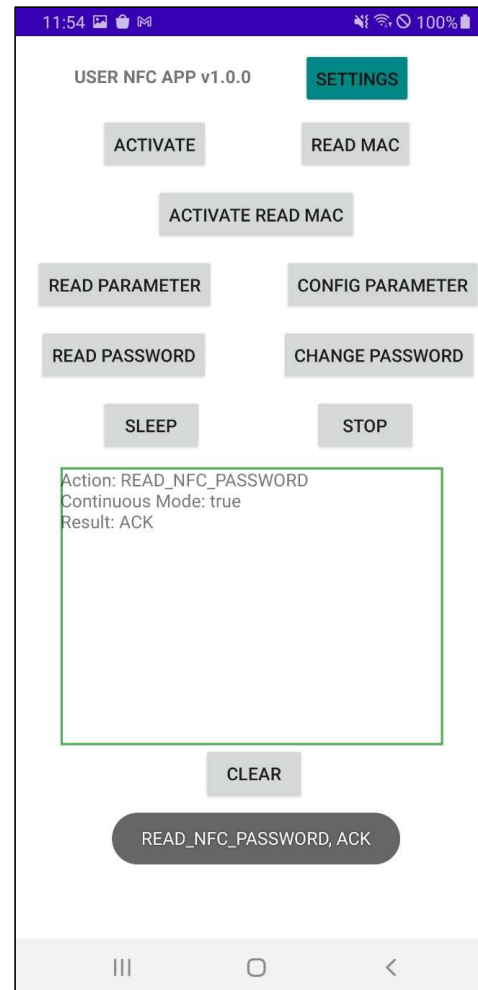
- The main page
 - The configuration result from the NFC Toolbox Service App
 - Click the “Read Parameter” button to verify parameters after the configuration



Sample User App Function

Read Beacon NFC password

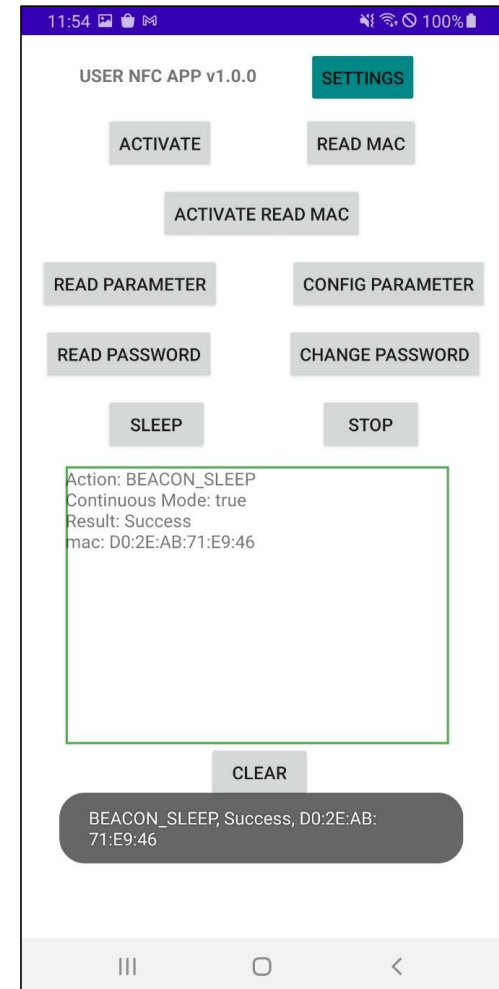
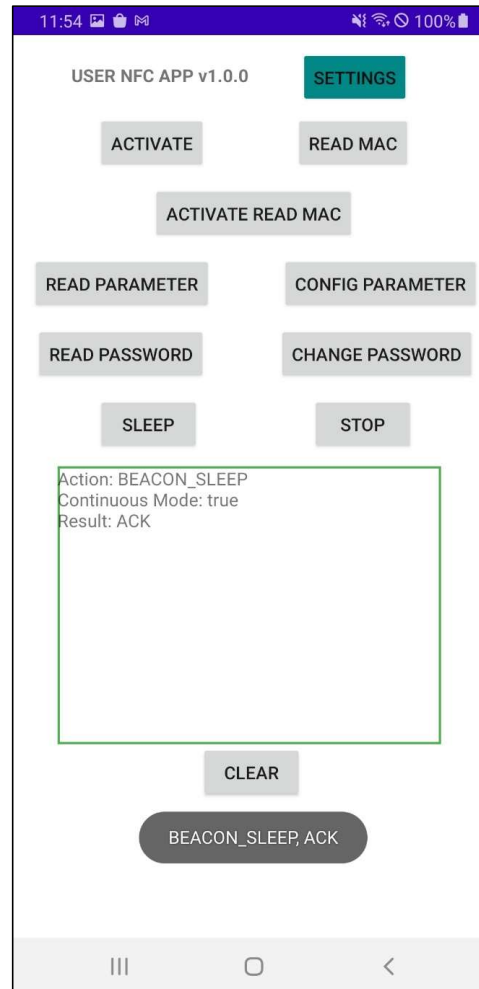
- The main page
 - Click the “Read Password” button
- The result from the NFC Toolbox Service App



Sample User App Function

Put the Beacon to sleep mode

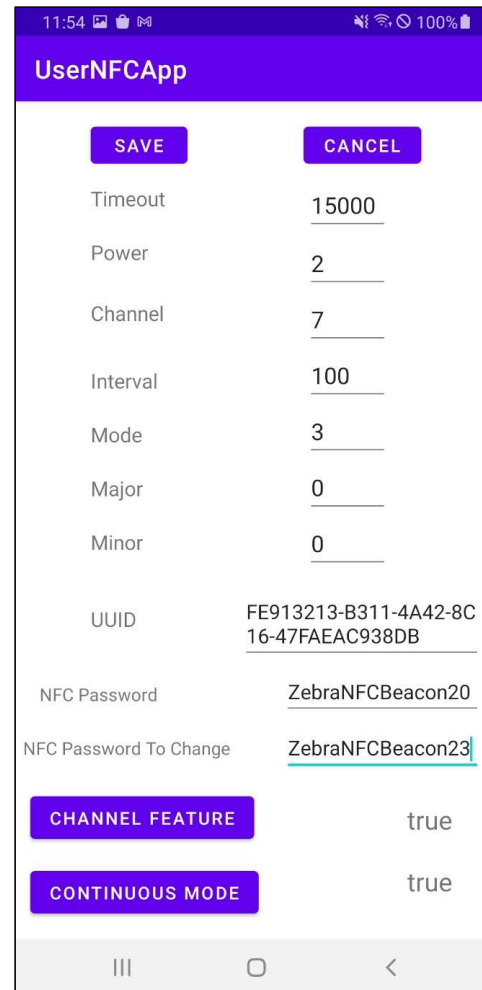
- The main page
 - Click the “Sleep” button
- The result from the NFC Toolbox Service App



Sample User App Function

Change Beacon NFC password

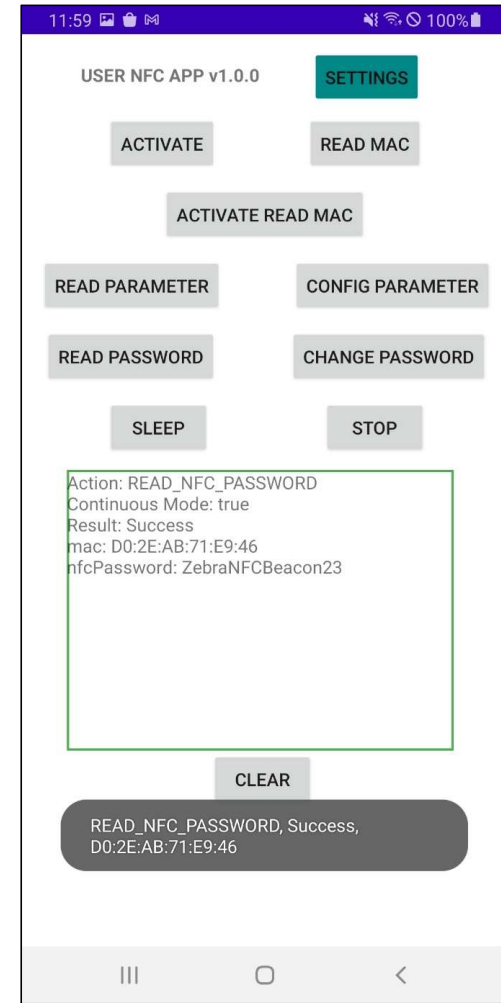
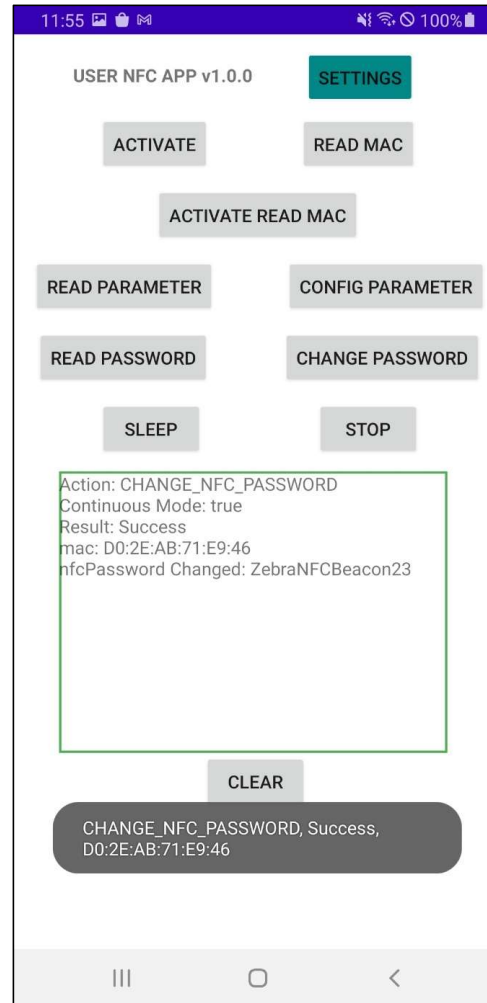
- The Settings page
 - Change the “NFC Password To Change” field
- The main page
 - Click the “Change Password” button



Sample User App Function

Change Beacon NFC password

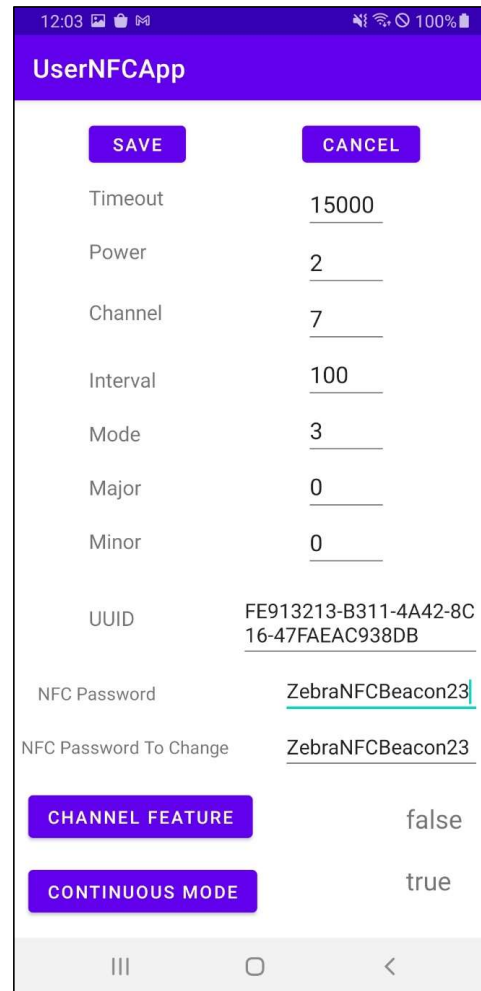
- The main page
 - The configuration result from the NFC Toolbox Service App
 - Click the “Read Password” button to verify NFC password after the configuration



Sample User App Function

After changing NFC password

- The Settings page
 - Need change the “NFC Password” field to be the same as “NFC Password To Change” field
- The main page
 - Click the “Read Parameter” button to read the beacon parameters to make sure the new NFC password is in effect



12:03 100%

UserNFCApp

SAVE CANCEL

Timeout 15000

Power 2

Channel 7

Interval 100

Mode 3

Major 0

Minor 0

UUID FE913213-B311-4A42-8C16-47FAEAC938DB

NFC Password ZebraNFCBeacon23

NFC Password To Change ZebraNFCBeacon23

CHANNEL FEATURE false

CONTINUOUS MODE true



11:53 100%

USER NFC APP v1.0.0 SETTINGS

ACTIVATE READ MAC

ACTIVATE READ MAC

READ PARAMETER CONFIG PARAMETER

READ PASSWORD CHANGE PASSWORD

SLEEP STOP

Action: READ_BEACON_PARAMETERS
Continuous Mode: true
Result: Success
mac: D0:2E:AB:71:E9:46
power: 2
mode: 3
channel: 7
interval: 100
major: 0
minor: 0
channelFeature: true
uuid: FE913213-B311-4A42-8C16-47FAEAC938DB

CLEAR

READ_BEACON_PARAMETERS, Success, D0:2E:AB:71:E9:46

Sample User App Function

Stop the NFC process

- The main page
 - Click the “Stop” button
- The result from the NFC Toolbox Service App



Thanks...



Rich Woodburn
Zebra Technologies
V: 917 780 2378 @zebra.com
M: 408-693-7199

