Newland Android PDA

UHF Module Configuration Developer Handbook

Revision History

Version	Description	Date
V1.0.0	Initial release	June 24, 2019

Table of Contents

About This Manual

This manual is developer guide to call PDA UHF parameter setting interfaces. Developers may refer to this handbook for configuration details.

Set Interface

Set UHF module parameters like inventory parameters, antenna power parameters, and extra parameters. Reading and writing tag performance vary with different parameters.

Interface definition

Class: com.nlscan.android.uhf.UHFManager	
Return Value	Method
static UHFManager	**getInstance()** To get instance
UHFReader.READER_STATE	**setParam(String paramKey,** **String paramName,** **String sValue)** Parameters setting Parameter: paramKey function identifier (Please refer to corresponding model for parameter identifier.) Parameter: paramName parameter name (Please refer to corresponding model for parameter identifier.) Parameter: sValue, Return value: (@See **UHFReader.READER_STATE**)
Map<String ,Object>	**getAllParams()** To get all parameters from returned Map. Return value: Map<String ,Object>

Example:

//get UHFManager instance

```
UHFManager mUHFMgr = UHFManager.getInstance();
```

//----set antenna power

```
JSONArray jsAntArray = new JSONArray();
try {
        JSONObject jobj = new JSONObject();
        jobj.put("antid", 1);//antenna ID
        jobj.put("readPower", 2700);//read power
        jobj.put("writePower", 2000);//write power
        jsAntArray.put(jobj);
} catch (Exception e) {
}
String sAntPowerValue = jsAntArray.toString();
String paramKey = "RF_ANTPOWER";
String paramName = "PARAM_RF_ANTPOWER";
UHFReader.READER_STATE er = mUHFMgr.setParam(paramKey, paramName, sAntPowerValue);
if( er == UHFReader.READER_STATE. OK_ERR)
        //success......
else
        //fail.....
```

//----get antenna power

```
//load “reader transmit power JSONArray”
//[{"antid":1,"readPower":2600,"writePower":2700},...] format
Map<String,Object> settingsMap = mUHFMgr.getAllParams();
String key = "RF_ANTPOWER";
String sJsonValue = (String) settingsMap.get(key);
JSONArray jsArray = new JSONArray(sValue);
int len = jsArray.length();
for(int i =0 ;i < len;i++)
{
        JSONObject jobj = jsArray.optJSONObject(i);
        int antid = jobj.optInt("antid");// Antenna ID
        short readPower = (short)jobj.optInt("readPower");// read power
        short writePower = (short)jobj.optInt("writePower");// write power
}
```

Parameters

1. Inventory

Inventory protocol:

Name	paramKey	paramName	Value Type	Value

Inventory protocol	**TAG_INVPOTL**	**PARAM_TAG_INVPOTL**	Set value: STRING Get value: INT[]	NONE: 0 ISO180006B: 3 GEN2: 5 ISO180006B_UCODE: 6 IPX64: 7 IPX256: 8 It is available to select several values with comma “,” separated. Default: 5（GEN2）
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("TAG_INVPOTL", "PARAM_TAG_INVPOTL", "3,5");			
	Get current value: UHFManager mUHFMgr = UHFManager.getInstance(); Map<String,Object> settingsMap = mUHFMgr.getAllParams(); String key = "TAG_INVPOTL"; int[] protocls = (int[]) settingsMap.get(key);			

Antenna

Name	paramKey	paramName	Value Type	Value
Antenna	**ANTS**	**PARAM_ANTS_GROUP**	Set value: STRING Get value: INT[]	Antenna: 1:1 It is available to select several values with comma “,” separated. Default: 1
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("ANTS", "PARAM_ANTS_GROUP", "1");			
	Get current value: UHFManager mUHFMgr = UHFManager.getInstance(); Map<String,Object> settingsMap = mUHFMgr.getAllParams();			

	String key = "PARAM_ANTS_GROUP"; int[] ants = (int[]) settingsMap.get(key);

Inventory Duration and Interval

Name	paramKey	paramName	Value Type	Value
Inventory duration	**INV_TIME_OUT**	**PARAM_INV_TIME_OUT**	Set value: STRING Get value: LONG	Duration (ms) Default 50 ms
Inventory interval	**INV_INTERVAL**	**PARAM_INV_INTERVAL_TIME**	Set value: STRING Get value: LONG	Interval (ms) Default 0 ms
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("INV_TIME_OUT", "PARAM_INV_TIME_OUT", "50"); UHFReader.READER_STATE er = mUHFMgr.setParam("INV_INTERVAL", "PARAM_INV_INTERVAL_TIME", "0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "INV_TIME_OUT";
	long timeout = (long) settingsMap.get(key);
	
	key = "INV_INTERVAL";
	long interval = (long) settingsMap.get(key);			

2. Antenna Power

Antenna Detection

Name	paramKey	paramName	Value Type	Value

<table>
<tr><td rowspan="3">Antenna detection</td><td>READER_IS_CHK_ANT</td><td>PARAM_READER_IS_CHK_ANT</td><td>Set value:
STRING

Get value:
INT[]</td><td>Detection:1
Not detection: 0

Default: 1
(detection)</td></tr>
<tr><td colspan="4">Set parameters:
UHFManager mUHFMgr = UHFManager.getInstance();
UHFReader.READER_STATE er = mUHFMgr.setParam("READER_IS_CHK_ANT",
"PARAM_READER_IS_CHK_ANT",
"1");</td></tr>
<tr><td colspan="4">Get current value:
UHFManager mUHFMgr = UHFManager.getInstance();
Map<String,Object> settingsMap = mUHFMgr.getAllParams();
String key = "READER_IS_CHK_ANT";
int[] antsCheck = (int[]) settingsMap.get(key);
if(antsCheck[0] == 1) //detection
//...
else //not detection
//...</td></tr>
</table>

Antenna Power

Power	paramKey	paramName	Value Type	Value
Power	**RF_ANTPOWER**	**PARAM_RF_ANTPOWER**	Set value: STRING Get value: String	Set multiple antennas' power and you will get multiple JSON objects from string in JSONArray format. e.g.: [{ "antid":1, "readPower":2600, "writePower":2700 }, ...] antid antenna ID, default 1 readPower read power（INT） writePower write power（INT）

				Default value:
				[
				{
				"antid":1,
				"readPower":2700,
				"writePower":2000
				}
				]

Set parameters:

```
UHFManager mUHFMgr = UHFManager.getInstance();
SONArray jsAntArray = new JSONArray();
try {
        JSONObject jobj = new JSONObject();
        jobj.put("antid", 1);//Antenna ID
        jobj.put("readPower", 2700);//read power
        jobj.put("writePower", 2000);//write power
        jsAntArray.put(jobj);
} catch (Exception e) {
}
String sAntPowerValue = jsAntArray.toString();
String paramKey = "RF_ANTPOWER";
String paramName = "PARAM_RF_ANTPOWER";
UHFReader.READER_STATE er = mUHFMgr.setParam(paramKey,
  paramName, sAntPowerValue);
if( er == UHFReader.READER_STATE. OK_ERR)
        //success.....
else
        //fail.....
```

Get current value:

```
UHFManager mUHFMgr = UHFManager.getInstance();
Map<String,Object> settingsMap = mUHFMgr.getAllParams();
String key = "RF_ANTPOWER";
String sJsonValue = (String) settingsMap.get(key);
JSONArray jsArray = new JSONArray(sValue);
int len = jsArray.length();
for(int i =0 ;i < len;i++)
{
        JSONObject jobj = jsArray.optJSONObject(i);
        int antid = jobj.optInt("antid");//Antenna ID
```

	short readPower = (short)jobj.optInt("readPower");// read power
	short writePower = (short)jobj.optInt("writePower");//write power
	}

Antenna Power under Low Battery

Name	paramKey	paramName	Value Type	Value
Enable/Disable	**LOWER_POWER**	**PARAM_LOWER_POWER_DM_ENABLE**	Set value: STRING Get value: INT	Enable: 1 Disable: 0 Default: 0 (Disable)
Battery Level	**LOWER_POWER**	**PARAM_LOWER_POWER_LEVEL**	Set value: STRING Get value: INT	Range: 0，10，15，20，25，30，35，40，45，50，60 Default:20（20%）
Power	**LOWER_POWER**	**PARAM_LOWER_POWER_DBM**	Set value: STRING Get value: INT	Range: 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200, 2300, 2400, 2500, 2600, 2700, 2800, 2900, 3000

				Default: 2000
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("LOWER_POWER", "PARAM_LOWER_POWER_DM_ENABLE", "1"); er = mUHFMgr.setParam("LOWER_POWER", "PARAM_LOWER_POWER_LEVEL", "25");//25% er = mUHFMgr.setParam("LOWER_POWER", "PARAM_LOWER_POWER_DBM", "2100");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "PARAM_LOWER_POWER_DM_ENABLE";
	int lowPowerEnable = (int)settings.get(key);
	if(lowPowerEnable == 1) //enable
	//...
	else //disable
	//...
	
	//battery level
	key = "PARAM_LOWER_POWER_LEVEL";
	int lowBatteryLevel = (int)settings.get(key);
	
	//power under low battey
	key = "PARAM_LOWER_POWER_DBM";
	int lowPowerDBM= (int)settings.get(key);			

3. Region Frequency

Region

Name	paramKey	paramName	Value Type	Value
Region	**FREQUENCY_REGION**	**PARAM_FREQUENCY_REGION**	Set value: STRING	China: 6 China 2: 10 North America: 1

			Get value: INT	Europe: 8 All: 255 Default: 1 (North America)
Frequency	**FREQUENCY_HOPTABLE**	**PARAM_HTB**	Set value: STRING Get value: INT[]	Obtain available range from region you set. e.g.: North America: 902750 903250 903750 ... It is available to select several values with comma “,” separated. e.g.: “902750,903250,903750, ...”

	Set parameters:
	UHFManager mUHFMgr = UHFManager.getInstance();
	//region:
	int RG_NA = 1 ;
	UHFReader.READER_STATE er = mUHFMgr.setParam("FREQUENCY_REGION",
	"PARAM_FREQUENCY_REGION",
	String.valueOf(RG_NA));
	//frequency:
	String shtb = "902750,903250,903750";
	er = mUHFMgr.setParam("FREQUENCY_HOPTABLE",
	"PARAM_HTB",
	shtb);			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	//region
	String key = "FREQUENCY_REGION";
	int region = (int)settingsMap.get(key);
	
	//fequency
	String key = "FREQUENCY_HOPTABLE";			

	int[] htbs = (int[])settingsMap.get(key);

4. Gen2

Session

Name	paramKey	paramName	Value Type	Value
Session	**POTL_GEN2_SESSION**	**PARAM_POTL_GEN2_SESSION**	Set value: STRING Get value: INT[] The 1st value	0: S0 1: S1 2: S2 3: S3 Default: 0（S0）
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("POTL_GEN2_SESSION", "PARAM_POTL_GEN2_SESSION", "0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "POTL_GEN2_SESSION";
	int[] sessions = (int[]) settingsMap.get(key);
	int ses = sessions[0];			

Q Value

Name	paramKey	paramName	Value Type	Value
Q value	**POTL_GEN2_Q**	**PARAM_POTL_GEN2_Q**	Set value: STRING Get value: INT[] The 1st value	Value: -1 ~ 14 Default: -1（Auto）
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("POTL_GEN2_Q",			

	"PARAM_POTL_GEN2_Q", "-1");

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "POTL_GEN2_Q";
	int[] arr = (int[]) settingsMap.get(key);
	int gen2Q = arr[0];

Write Mode

Name	paramKey	paramName	Value Type	Value
Write mode	**POTL_GEN2_WRITEMODE**	**PARAM_POTL_GEN2_WRITEMODE**	Set value: STRING Get value: INT[] The 1st value	0: word write 1: block write Default: 0 (word write)
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("POTL_GEN2_WRITEMODE", "PARAM_POTL_GEN2_WRITEMODE", "0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "POTL_GEN2_WRITEMODE";
	int[] arr = (int[]) settingsMap.get(key);
	int writeMode = arr[0];			

Max Length

Name	paramKey	paramName	Value Type	Value
Write mode	**POTL_GEN2_MAXEPCLEN**	**PARAM_POTL_GEN2_MAXEPCLEN**	Set value: STRING Get value:	(bit address) 96 496

			INT[] The 1st value	Default: 496
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("POTL_GEN2_MAXEPCLEN", "PARAM_POTL_GEN2_MAXEPCLEN", "96");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "POTL_GEN2_MAXEPCLEN";
	int[] arr = (int[]) settingsMap.get(key);
	int maxLen = arr[0];			

Target

Name	paramKey	paramName	Value Type	Value
Write mode	**POTL_GEN2_TARGET**	**PARAM_POTL_GEN2_TARGET**	Set value: STRING Get value: INT[] The 1st value	0: A 1: B 2: A-B 3: B-A Defaut: 0 (A)
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("POTL_GEN2_TARGET", "PARAM_POTL_GEN2_TARGET", "0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "POTL_GEN2_TARGET";
	int[] arr = (int[]) settingsMap.get(key);
	int target = arr[0];			

Gen2 Tag Encoding

Name	paramKey	paramName	Value Type	Value

Write mode	**POTL_GEN2_TAGENCODING**	**PARAM_POTL_GEN2_TAGENCODING**	Set value: STRING Get value: INT[] The 1st value	0: FM0 1: M2 2: M4 3: M8 Default: 2 (M4)
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("POTL_GEN2_TAGENCODING", "PARAM_POTL_GEN2_TAGENCODING", "0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "POTL_GEN2_TAGENCODING";
	int[] arr = (int[]) settingsMap.get(key);
	int target = arr[0];			

5. Filter

Name	paramKey	paramName	Value Type	Value

Filter	**TAG_FILTER**	**PARAM_TAG_FILTER**	Set value:
			STRING
			
			Get value:
			STRING
				String in JSON format
				e.g.:
				{
				"fdata":"0A0B0D0E0F",
				"flen":"40",
				"bank":"1".
				"startaddr":"32",
				"isInvert":"1"
				}
				
				Field explanation:
				Fdata: Filter data (hexadecimal string）
				Flen: Filter data length (Need to convert to byte）
				Bank: Filter data memory bank(e.g. EPC bank)
				Startaddr: Start address

				（bit address） isInvert: Match/unmatch (0-match, 1-unmatch）

	Set parameters:
	UHFManager mUHFMgr = UHFManager.getInstance();
	JSONObject jsItem = new JSONObject();
	int bank = 1;//EPC bank
	String sHexFdata = "0A0B0D0E0F";
	int ln=sHexFdata.length();
	if(ln==1||ln%2==1)
	ln++;
	int flen = (ln/2)*8;
	int startaddr = 32;
	int isInvert = 0;
	jsItem.put("bank", bank);
	jsItem.put("fdata", sHexFdata);
	jsItem.put("flen", flen);
	jsItem.put("startaddr",startaddr);
	jsItem.put("isInvert", isInvert);
	String sValue = jsItem.toString();
	//set filterconditions
	UHFReader.READER_STATE er = mUHFMgr.setParam("TAG_FILTER",
	"PARAM_TAG_FILTER",
	sValue);
	//remove filter conditions
	er = mUHFMgr.setParam("TAG_FILTER",
	"PARAM_TAG_FILTER",
	null);			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "TAG_FILTER";
	String sValue = (String) settingsMap.get(key);
	JSONObject jsItem = new JSONObject(sValue);
	int bank = jsItem.optInt("bank");
	String sHexFdata = jsItem.optString("fdata");
	int flen = jsItem.optInt("flen");
	int startaddr = jsItem.optInt("startaddr");
	int isInvert = jsItem.optInt("isInvert");			

6. CustomData

<table>
<tr><th>Name</th><th>paramKey</th><th>paramName</th><th>Value Type</th><th>Value</th></tr>
<tr><td rowspan="2">Inventory protocol</td><td>TAG_EMBEDEDDATA</td><td>PARAM_TAG_EMBEDEDDATA</td><td>Set value:
STRING

Get value:
STRING</td><td>String in JSON format
e.g.:
{
"startaddr":"2",
"bytecnt":"4",
"bank":"1".
"accesspwd":"00000001"
}

Field explanation:
Startaddr: Start address (unit: block, that is 16-bit address)
bytecnt : Number of bytes
bank: memory bank (e.g. EPC bank)
accesspwd: Access password (hexadecimal string, you don’t need to set this item if access password is not set.)</td></tr>
<tr><td colspan="4">Set parameters:
UHFManager mUHFMgr = UHFManager.getInstance();
int bank = 1;//EPC
int startaddr = 2;//start address (block)
int bytecnt = 4;
String sHexAccesspwd = "00000001";//hexadecimal string
JSONObject jsItem = new JSONObject();
jsItem.put("bank", bank);
jsItem.put("startaddr", startaddr);
jsItem.put("bytecnt", bytecnt);
jsItem.put("accesspwd", sHexAccesspwd);
String sValue = jsItem.toString();
//set CustomData
UHFReader.READER_STATE er = mUHFMgr.setParam("TAG_EMBEDEDDATA",
"PARAM_TAG_EMBEDEDDATA",
sValue);
//remove CustomData
er = mUHFMgr.setParam("TAG_EMBEDEDDATA",</td></tr>
</table>

	"PARAM_TAG_EMBEDEDDATA", null);

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "TAG_EMBEDEDDATA";
	String sValue = (String) settingsMap.get(key);
	JSONObject jsItem = new JSONObject(sValue);
	int bank = jsItem.optInt("bank");
	String sHexAccesspwd = jsItem.optString("accesspwd");
	int bytecnt = jsItem.optInt("bytecnt");
	int startaddr = jsItem.optInt("startaddr");

7. Other parameters

Antenna Singulation

Name	paramKey	paramName	Value Type	Value
Antenna singulation	**TAGDATA_UNIQUEBYANT**	**PARAM_TAGDATA_UNIQUEBYANT**	Set value: STRING Get value: INT[]	1: Unique 0：Not unique Default: 0 (Not unique)
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("TAGDATA_UNIQUEBYANT", "PARAM_TAGDATA_UNIQUEBYANT", "0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "TAGDATA_UNIQUEBYANT";
	int[] arr = (int[]) settingsMap.get(key);
	int unique = arr[0];			

Custom Data Singulation

Name	paramKey	paramName	Value Type	Value
Inventory protocol	**TAGDATA_UNIQUEBYEMDDATA**	**PARAM_TAGDATA_UNIQUEBYEMDDATA**	Set value: STRING	1: Unique 0：Not unique

			Get value: INT[]	Default: 0 (Not unique)
	Set parameters: UHFManager mUHFMgr = UHFManager.getInstance(); UHFReader.READER_STATE er = mUHFMgr.setParam("TAGDATA_UNIQUEBYEMDDATA", "PARAM_TAGDATA_UNIQUEBYEMDDATA", "0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "TAGDATA_UNIQUEBYEMDDATA";
	int[] arr = (int[]) settingsMap.get(key);
	int unique = arr[0];			

Temperature

Name	paramKey	paramName	Value Type	Value
Temperature	**TEMPTURE**	**PARAM_TEMPTURE**	Read only Get valute: INT	Module temperature (℃)

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	String sValue = mUHFMgr.getParam("TEMPTURE", "PARAM_TEMPTURE", null);
	if(sValue != null && TextUtils.isDigitsOnly(sValue))
	int tempture = Integer.parseInt(sValue);			

8. Rapid Mode

Name	paramKey	paramName	Value Type	Value
Inventory protocol	**INV_QUICK_MODE**	**PARAM_INV_QUICK_MODE**	Set value: STRING Get value: INT[]	Enable: 1 Disable: 0 Default: 0 (Disable)

	POTL_GEN2_SESSION	**PARAM_POTL_GEN2_SESSION**	Set value: STRING Get value: INT[] The 1st value	GEN2 Session S1: 1 (for rapidly inventorying a large number of tags) S0：0 (for rapidly inventorying a small number of tags) Default: S0

	Set parameters:
	UHFManager mUHFMgr = UHFManager.getInstance();
	//enable rapid mode
	UHFReader.READER_STATE er = mUHFMgr.setParam("INV_QUICK_MODE",
	"PARAM_INV_QUICK_MODE",
	"1");
	//enable GEN2 Session, choose only S1 or S0.
	//enable GEN2 Session : S1
	er = mUHFMgr.setParam("POTL_GEN2_SESSION",
	"PARAM_POTL_GEN2_SESSION",
	"1");
	//enable GEN2 Session : S0
	er = mUHFMgr.setParam("POTL_GEN2_SESSION",
	"PARAM_POTL_GEN2_SESSION",
	"0");			

	Get current value:
	UHFManager mUHFMgr = UHFManager.getInstance();
	Map<String,Object> settingsMap = mUHFMgr.getAllParams();
	String key = "INV_QUICK_MODE";
	int quickModeEnable = (int) settingsMap.get(key);
	
	key = "POTL_GEN2_SESSION";
	int[] sessions = (int[]) settingsMap.get(key);
	int ses = sessions[0];			

Newland Auto-ID Tech. Co., Ltd.
(Headquarters)
3F, Building A, No.1, Rujiang West Rd., Mawei, Fuzhou, Fujian, China 350015
Tel: +86 - (0) 591-83978605
Fax: +86 - (0) 591-83979216
E-mail: contact@nlscan.com
Web: www.newlandaidc.com

Newland Europe BV
Rolweg 25, 4104 AV Culemborg, The Netherlands
Tel: +31 (0) 345 87 00 33
Fax: +31 (0) 345 87 00 39
Email: info@newland-id.com
Web: www.newland-id.com
Tech Support: tech-support@newland-id.com

Newland North America Inc.
46559 Fremont Blvd., Fremont, CA 94538, USA
Tel: 510 490 3888
Fax: 510 490 3887
Email: info@newlandna.com
Web: www.newlandamerica.com

Newland Latin America
Tel: +1 (239) 598 0068
Fax: +1 (239) 280 1238
Email: info@newlandla.com
Web: www.newlandamerica.com

Newland Taiwan Inc.
7F-6, No. 268, Liancheng Rd., Jhonghe Dist. 235, New Taipei City, Taiwan
Tel: +886 2 7731 5388
Fax: +886 2 7731 5389
Email: info@newland-id.com.tw
Web: www.newland-id.com.tw

Newland Korea
Biz. Center Best-one, Jang-eun
Medical Plaza 6F, Bojeong-dong 1261-4,
Kihung-gu, Yongin-City, Kyunggi-do, South Korea
Tel: +82 10 8990 4838
Fax: +82 70 4369 0009
Email: th.sung@newland-id.com.tw
Web: www.newlandaidc.com/kor/