TIRIS Bluetooth PUK Programming Guide

Technology Solutions UK Ltd Suite C,
Loughborough Technology Centre,
Epinal Way,
Loughborough,
Leicestershire,
LE11 3GE,
United Kingdom.

Tel: +44 (0) 1509 238248
Fax: +44 (0) 1509 220020

Email: enquiries@tsl.uk.com

Contents

History

Version	Date	Modifications
1.0	16/06/04	Document Creation
1.1	15/8/2004	Minor corrections
1.2	22/10/04	Added configuration and power save commands, added description of power save features

1 Introduction

This guide is for programmers wishing to communicate with the TIRIS Bluetooth PUK. It defines the packet structure to and from the TIRIS Bluetooth PUK, the error packets and the commands.

Communication between the TIRIS Bluetooth PUK and the host computer uses Bluetooth. The PUK presents a comm. port as a Bluetooth service. This runs at 57600 baud with 8 data bits, no parity and 1 stop bit

2 Using the PUK

2.1 Turning the PUK on and off

To activate the PUK, press and release the power button. To turn the PUK off press and hold the power button for 3 seconds. The PUK will auto power off after 60 seconds in the absence of a Bluetooth connection (unless a different value has been configured).

2.2 Configuring the PUK

Three aspects of the PUK operation can be configured. These are the power down delay, the button action and the Bluetooth power saving modes.

The power down delay is the time taken for the PUK to switch off when the Bluetooth link is not connected. This delay is seen in two situations; the length of time the PUK stays powered after a button press if it is not connected to and the length of time the PUK stays powered after the host has disconnected from it. Long power down delays may be required in some situations, but the delay should not be increased more than necessary. During the power down delay period the PUK draws a relatively high current and so the battery life may be reduced if the PUK is left in this state for a long time.

The PUK can be programmed so that it will read transponders when the button is pressed. Any transponders read will be sent to the host in ASCII with a carriage return and line feed terminator. The number of times the PUK will perform this read cycle can also be programmed. If this mode is enabled there is no need for the host to issue any commands to the PUK.

The Bluetooth power saving modes are described in Section 2.4

Three configuration bytes are used to configure the PUK. The command used to program the configuration bytes is described in Section 5.6.

2.3 Charging

To charge the PUK connect the PUK to the power supply. The light on the PUK indicates the charging state red for fast charging, green for charged.

Please note:-

There is a safety feature built into the PUK battery such that if the battery is very flat it will shut down and may require several hours charging to recover to a state where a fast charge is allowed. This should only occur if the PUK has been left for a long period unused. Under normal operation the PUK will shut off before the battery gets to this state. If you are not using the PUK for long periods please fit the transport cap provided.

2.4 Power saving

The PUK has a power save mode which can be turned permanently on or off by command. Alternatively the PUK can be left to automatically enter power save mode. Using the power saving modes will provide a significant increase in operating time for the PUK. The transponder RF section has its own auto power save mode. This is always enabled.

2.4.1 Power Save Mode

When the power save is active there will be a delay of up to 500ms for a command to be processed. In most cases this is not a problem and power save should be permanently enabled to provide maximum battery life. Power saving works by using the standard Bluetooth sniff mode. In this mode the Bluetooth link only communicates every 500ms. Once data has been transferred the link remains active for 200ms. This means that subsequent commands will not be delayed so long as there is not a gap of more than 200ms between subsequent commands.

Please note that the host Bluetooth device must support sniff mode for power saving to work.

2.4.2 Auto Power Save

The auto power save mode offers a compromise between power consumption and speed of response. It works by taking the unit out of power save mode whenever a command is received and then returns to power save mode after about 4 seconds in the absence of further commands. The auto mode works well where there are gaps over 5 seconds between groups of commands.

2.4.3 Transponder RF Section Power Save

The power to the RF section is turned off if there is no communication from the host for 3 Seconds. The RF section is powered up whenever a communication is received from the host.

3 Packet Structure

The packet structure will be the same for both outgoing (command, host to PUK) and incoming (response, PUK to host). It will consist of the following fields:

- Start of Packet (02_{hex})
- RFU (byte reserved for future use, should be zero, 00_{hex})
- Destination byte (Dest)
- Command byte (CMD)
- Options byte (Opt)
- Length (least significant byte) of data section = nl
- Length (most significant byte) of data section = nh
- Parameter section (nh x 256 + nl bytes total)
- Checksum (CS_1, CS_2)

This is illustrated in the diagram below:

Figure 1. General Packet Structure

SOP	**RFU**	**DST**	**CMD**	**Opt**	**Length (LSB)**	**Length (MSB)**	**P_0**	**P_1**		**P_k**	**CS_{lsb}**	**CS_{msb}**
02_{hex}	00_{hex}				nl	nh						

Note. k = nh x 256 +nl-1

3.1 Start of Packet

The start of packet allows the PUK to identify the start of a packet and is always 02_{hex}.

3.2 RFU

This byte is reserved for future use and should always be 00_{hex}.

3.3 Destination

The destination byte is used to determine where the subsequent command byte is destined for.

e.g.

Destination 01hex is the PUK itself and the command might be to get the PUK version information.

Destination 03hex is a TIRIS transponder in the PUK's reading zone and the command might be to read a page of data.

3.4 Command

The Command byte specifies the action to be performed and must be appropriate for the destination specified.

3.5 Length

The length bytes combine to form the length of the parameter section, least significant byte first. Although this structure allows for a maximum length parameter section of over sixty thousand bytes this will typically be limited by the PUK. An invalid length error will be sent if the length specified is greater than the interface can receive.

3.6 Options

The options byte is a set of eight flags that can be set or cleared. The function of each flag is specific to the command and destination byte used. In the response the option byte should only have one of two values; 00_{hex} for a successful transaction or FF_{hex} if an error has occurred. Error Packets and Error Codes are discussed in the next section of this document.

3.7 Parameter Section

The parameter section holds data appropriate to the specific incoming or outgoing packet. The length of this section in bytes is specified by the length section.

3.8 Checksum

The 16 bit checksum (LSB first) is calculated by the addition of all the preceding bytes. If the value exceeds 16 bits the result is truncated to the 16 least significant bits which form the checksum.

4 The Error Packet

The PUK reports errors using the standard packet structure. A value of FF_{hex} in the option byte is used to indicate that an error has occurred. The first parameter byte contains the error code and depending on the error there may be further bytes in the parameter section giving more information about the error. The packet structure is shown below.

SOF	RFU	DST	CMD	Opt	Length (LSB)	Length (MSB)	P_0	P_1		P_k	CS_{lsb}	CS_{msb}
02_{hex}	00_{hex}			FF_{hex}			Error Code					

Figure 2. Error response packet

Note: The error response may be sent before the full transmission of an outgoing packet; for example an invalid length, greater than can be handled by the PUK, will be sent immediately after the second length byte.

4.1 Error Codes

Error Code	Description
01_{hex}	Transponder Generated Error
02_{hex}	Destination Not Recognised
03_{hex}	Command Not Recognised
04_{hex}	Invalid options
05_{hex}	Invalid length
06_{hex}	Invalid Checksum
07_{hex}	No Transponder Present
08_{hex}	Invalid Parameters
20_{hex}	Write serial number failed
$E0_{hex}$ to EF_{hex}	Bootloader error
FF_{hex}	Undefined error

4.2 Transponder Generated Error (01_{hex})

The PUK has successfully received the request and the requested action has been sent to the Transponder. However the Transponder has responded with an error. The rest of the parameter section will contain the error response from the Transponder. For example this error could be caused by an attempt to write to a locked block.

4.3 Destination Not Recognised (02_{hex})

The PUK does not recognise the destination byte received as a destination for which it can generate a suitable response.

4.4 Command Not Recognised (03_{hex})

The PUK does not recognise the command byte received as a command for which it can generate a suitable response.

4.5 Invalid Options (04_{hex})

The options specified are not appropriate for the specified command and destination.

4.6 Invalid Length (05_{hex})

Typically the PUK has received a start of packet and length bytes but failed to receive sufficient bytes to fulfil the length requirement before a serial timeout occurred. Alternatively the length specified may be too great for the PUK to handle.

4.7 Invalid Checksum (06_{hex})

The packet has been received up to one of the checksum bytes at which point the calculated checksum did not match one of the received checksum bytes.

4.8 No Transponder Present (07_{hex})

This error is generated when the PUK has failed to find a suitable transponder in the antenna field to perform the requested command on.

4.9 Invalid Parameters (08_{hex})

The Parameters specified are not appropriate for the specified command and destination.

4.10 Bootloader Errors ($E0_{hex}$ to EF_{hex})

This range of error codes can only be generated when upgrading the firmware and are handled by the firmware update software.

4.11 Undefined Error (FF_{hex})

This error code represents an unspecified error.

5 PUK Commands Destination 01hex

All the commands that control the PUK are supported using a command destination byte value of 01_{hex}.

To aid clarity the example packets have the parameter section greyed.

5.1 Option Byte Flags for PUK commands

The option byte contains one flag which is used by the PUK commands.

Bit 0 is reserved for future use and should be set to 0.

Bit 1 is used by the carrier on/off command. If set the LF carrier is turned on and if clear then the LF carrier is turned off.

Bits 2 to 7 are reserved for future use and should be set to 0.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	LF Carrier Flag	0

5.2 PUK commands

Below is a list of all the available Destination 01_{hex} PUK commands

Command	Description	Applicable Option Byte flags
01_{hex}	Read Version	None
04_{hex}	Read Serial Number	None
10_{hex}	Carrier on/off	LF Carrier Flag
$F0_{hex}$	Write Serial Number	None
$F1_{hex}$	Reset	None

5.3 Command 01_{hex} (Read Version)

This command is used to get the version number of the firmware and loader program in the PUK.

5.3.1 Request packet parameter section

This command does not require any parameters.

Example:

Request Version information from the PUK

Request Packet:

02 00 01 01 00 00 00 04 00

5.3.2 Response packet parameter section

Six Byte Response	The first 3 bytes represent the version number of the firmware loaded into the PUK LSB first. The next 3 bytes represent the version number of the loader program that downloads new firmware into the PUK LSB first.
Three Byte Response	These bytes represent the version number of the loader program that downloads new firmware into the PUK LSB first.

A three byte response is only obtained if firmware is not present in the PUK or if control has been transferred to the loader program by initialising the PUK for firmware downloading.

Example:

Response Packet:

02 00 01 01 00 06 00 00 02 01 01 00 01 0F 00

Firmware Version	1.2.0
Loader Version	1.0.1

5.4 Command 04_{hex} (Read Serial Number)

This command is used to read the 64 byte serial number of the PUK.

5.4.1 Request packet parameter section

This command does not require any parameters.

Example:

Request the serial number from the PUK

Request Packet:

02 00 01 04 00 00 00 07 00

5.4.2 Response packet parameter section

Serial Number (64 bytes) The 64 bytes represent the serial number of the PUK LSB first in ASCII text.

Example:

Response Packet:

02 00 01 04 00 40 00 41 30 31 32 33 34 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C4 01

The Serial number is “B43210A”

5.5 Command 10_{hex} (Carrier On/Off)

This command is used to turn the RF Carrier signal on or off.

The LF carrier flag in the option byte is used to signal if the carrier is to be turned on (Flag set) or to be turned off (Flag clear).

5.5.1 Request packet parameter section

This command does not require any parameters.

Example:

Turn the LF carrier on.

Request Packet:

02 00 01 10 02 00 00 15 00

5.5.2 Response packet parameter section

A single byte is returned in the parameter section with a value of zero if successful.

Example:

Response Packet:

02 00 01 10 02 01 00 00 16 00

Command successful.

5.6 Command B0$_{hex}$ (Program Configuration Bytes)

This command is used to program the configuration bytes. 3 bytes are used on the TIRIS Bluetooth PUK.

5.6.1 Request packet parameter section

The parameter section contains the 3 configuration bytes.

Example:

Program the configuration bytes with 0F 81 06$_{hex}$

Request Packet:

02 00 01 B0 00 03 00 0F 81 06 4C 01

5.6.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 B0 00 00 00 B3 00

Command successful.

Note that if the configuration bytes are programmed to make the PUK return transponder UIDs on button press then responses from the PUK are in ASCII format terminated by Carriage Return Line Feed.

5.6.3 Configuration Byte 1

This byte determines the number of read cycles that are performed after the button is pressed. By setting this byte to zero the PUK will read continually. The operations in a read cycle are defined by the second configuration byte. The read can be cancelled at any time by a second press of the button for 1 second.

5.6.4 Configuration Byte 2

The second configuration byte is a set of eight flags that can be set or cleared. The function of each flag is detailed below:-

Bit 0: Setting this bit allows TIRIS Transponders to read when the button is pressed.

Bits 1 to 4: Should be set to 0.

Bit 5: Setting this bit tells the unit to power up with power save off. Power saving can be turned on later by command, alternatively if the auto flag is set then the unit will use auto power save mode.

Bit 6: Setting this bit tells the unit to use auto power save. This bit will over-ride Bit 5. Setting both bits results will result in auto power save operation rather than power save off operation.

Bit 7: Setting this bit tells the unit to cancel the read cycles immediately a transponder has been successfully read.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Stop when read	Auto power save	Power save at Start up	0	0	0	0	TIRIS

5.6.5 Configuration Byte 3

This byte determines the number of ten second intervals that must elapse before the PUK automatically powers down if there is no Bluetooth connection from the host. The default value is 06_{hex} (60s).

5.6.6 Configuration Example

Read 15 times on button press, read TIRIS transponders, stop reading if transponder is found, power up in power save mode, do not use auto power save and use 60 seconds auto power down.

Configuration byte 1 = $0F_{hex}$

Configuration byte 2 = 81_{hex} ($1000\ 0001_{bin}$)

Configuration byte 3 = 06_{hex}

Request Packet to program the above configuration:

02 00 01 B0 00 03 00 0F 81 06 4C 01

5.7 Command $B2_{hex}$ (Power Save On)

This command is used to turn the power saving on.

5.7.1 Request packet parameter section

This command does not require any parameters.

Example:

Turn on the power saving.

Request Packet:

02 00 01 B2 00 00 00 B5 00

5.7.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 B2 00 00 00 B5 00

Command successful.

5.8 Command $B3_{hex}$ (Power Save Off)

This command is used to turn the power saving off.

5.8.1 Request packet parameter section

This command does not require any parameters.

Example:

Turn off the power saving.

Request Packet:

02 00 01 B3 00 00 00 B6 00

5.8.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 B3 00 00 00 B6 00

Command successful.

5.9 Command $F0_{hex}$ (Write Serial Number)

This command is used to write the 64 byte serial number to the PUK. The serial number is a write once option and is usually programmed at manufacture. Attempting to write a serial number a second time will generate an error response.

5.9.1 Request packet parameter section

Serial Number (64 bytes) The 64 bytes represent the serial number of the PUK LSB first.

Example:

Write the serial number “B43210A” to the PUK

Request Packet:

02 00 01 F0 00 40 00 41 30 31 32 33 34 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0 02

5.9.2 Response packet parameter section

There are no sections in the response parameter section.

Example:

Response Packet:

02 00 01 F0 00 00 00 F3 00

Write successful

Specific Error Codes

If the write fails error code 20_{HEX} is sent.

5.10 Command $F1_{hex}$ (Reset)

This command is used to reset the PUK. This is a complete reset including the Bluetooth interface. You may have to reconnect the Bluetooth link after this command.

5.10.1 Request packet parameter section

This command does not require any parameters.

Example:

Reset the PUK.

Request Packet:

02 00 01 F1 00 00 00 F4 00

5.10.2 Response packet parameter section

A single byte is returned in the parameter section with a value of zero to indicate that the reset is about to occur.

Example:

Response Packet:

02 00 01 F1 00 01 00 00 F5 00

Reset successful.

6 TIRIS^{tm} Commands Destination 03_{hex}

TIRIS^{tm} transponders are supported on the PUK using a command destination byte value of $03_{hex.}$

The reader is expected to be familiar with the TIRIS^{tm} transponders.

Full details of the TIRIS^{tm} transponder and the TIRIS^{tm} protocol are available from Texas Instruments (TI).

To aid clarity the example packets have the parameter section greyed.

6.1 Option Byte Flags for TIRIS^{tm} commands

The option byte is not used with commands for the TIRIS^{tm} transponders and should be set to zero.

6.2 TIRIS^{tm} Transponder Type Byte Values

Many of the TIRIS^{tm} commands respond with a byte representing the transponder type. The table below lists the possible values for this byte.

00_{hex}	Read Only Transponder
01_{hex}	Read/Write Transponder
02_{hex}	Multi-Page Transponder
03_{hex}	DST Transponder
04_{hex}	Selectable Addressable Multi-Page Transponder

6.3 TIRIS^{tm} Status byte and page values

Many of the TIRIS^{tm} commands respond with a page number and status byte. The table below lists the possible values for these bytes.

Page	Status	Description
1 to 17	00_{hex}	Read unlocked page
1 to 17	01_{hex}	Programming done
1 to 17	02_{hex}	Read locked page
1 to 17	03_{hex}	Reserved
0	00_{hex}	Read unlocked page, locking not correctly executed
0	01_{hex}	Programming done, but possibly not reliable
0	02_{hex}	Read locked page, but possibly not reliable

6.4 TIRIS^{tm} commands

Below is a list of all the available TIRIS^{tm} commands.

Command	Description
01_{hex}	Charge Only Read
02_{hex}	Write 64bit Read/Write Transponder
03_{hex}	General Page Read
04_{hex}	General Page Write
05_{hex}	General Page Lock
06_{hex}	Selective Page Read *1
07_{hex}	Selective Page Write *1
08_{hex}	Selective Page Lock *1
09_{hex}	DST Read *1
$0A_{hex}$	DST Write *1
$0B_{hex}$	DST Challenge Response *1

*1 Available in custom firmware only.

6.5 Command 01_{hex} (Charge Only Read)

This command is used to perform a charge only read. Most TIRIS[tm] transponder types respond to the command.

6.5.1 Request packet parameter section

There are no fields in the parameter section.

Example:

Perform a charge only read.

Request Packet:

02 00 03 01 00 00 00 06 00

6.5.2 Response packet parameter section

The first byte of the response parameter section is the transponder type byte. The rest of the response is transponder dependent. The table below shows the different responses.

Transponder Type	Response
Read Only Transponder	00_{hex}; 64 bit ID (LSB first)
Read/Write Transponder	01_{hex}; 64 bit read/write data (LSB first)
Multi-Page Transponder	02_{hex} ; Page number byte ; Status byte ; 64 bit page 1 data (LSB first)
DST Transponder	No response
SA Multi-Page Transponder	No response

Example:

Response Packet:

02 00 03 01 00 0B 00 02 01 02 11 22 33 44 55 66 77 88 7A 02

Transponder Type: 02_{hex} (Multi-Page)

Page number: 01_{hex}

Status: 02_{hex} (Locked Page)

Page 1 Data: 8877665544332211_{hex}

6.6 Command 02_{hex} (Write a Read/Write Transponder)

This command is used to write 64bits of data to the TIRIS™ Read/Write transponder.

6.6.1 Request packet parameter section

Data (8 bytes). The data to be written to the transponder LSB first.

Example:

Write data $FEDCBA9876543210_{HEX}$ to a TIRIS™ read/write transponder.

Request Packet:

02 00 03 02 00 08 00 10 32 54 76 98 BA DC FE 47 04

6.6.2 Response packet parameter section

Transponder Type (1 Byte).	This byte indicates the transponder type. For a successful write this should indicate a read/write transponder (01_{HEX}).
Transponder Data (8 Bytes).	This is the data read back from the transponder after programming. This should be verified to confirm correct programming.

Example:

Response Packet:

02 00 03 02 00 09 00 01 10 32 54 76 98 BA DC FE 49 04

Transponder Type:	01_{HEX} Read/Write
Data:	$1032547698BADCFE_{hex}$

The data matches that sent so the programming was successful.

6.7 Command 03_{hex} (General Page Read)

This command is used to read a page of data from a Multi-Page TIRIS™ transponder.

6.7.1 Request packet parameter section

Page Number (1 byte). The page number of the page to read.

Example:

Read page 4 of a TIRIS™ transponder.

Request Packet:

02 00 03 03 00 01 00 04 0D 00

6.7.2 Response packet parameter section

Transponder Type (1 Byte).	This byte indicates the transponder type. For a successful read this should indicate a Multi-Page transponder (02_{HEX}).
Page Number (1 Byte).	This should match the page number of the request. If you request a higher page than is available then the transponder will respond with the highest page available.
Status byte (1 Byte).	This byte indicates the status of the page.
Transponder Data (8 Bytes).	This is the data read back from the transponders page.

Example:

Response Packet:

02 00 03 03 00 0B 00 02 04 00 01 02 03 04 05 06 07 08 3D 00

Transponder Type: 02_{hex} (Multi-Page)

Page number: 04_{hex}

Status: 00_{hex} (Unlocked Page)

Page 4 Data: 0807060504030201_{hex}

6.8 Command 04_{hex} (General Page Write)

This command is used to write data to a page of a Multi-Page TIRIS™ transponder.

6.8.1 Request packet parameter section

Page Number (1 byte).	The page number of the page to write.
Page Data (8 Bytes).	This is the data to write to the transponders page.

Example:

Write page 6 of a TIRIS™ transponder with 7766554433221100_{HEX}.

Request Packet:

02 00 03 04 00 09 00 06 00 11 22 33 44 55 66 77 F4 01

6.8.2 Response packet parameter section

Transponder Type (1 Byte).	This byte indicates the transponder type. For a successful write this should indicate a Multi-Page transponder (02_{HEX}).
Page Number (1 Byte).	This should match the page number of the request.
Status byte (1 Byte).	This byte indicates the status of the page. For successful programming this should be 01_{hex} (Programming Done)
Transponder Data (8 Bytes).	This is the data read back from the transponders page after programming. This should be verified to confirm correct programming.

Example:

Response Packet:

02 00 03 04 00 0B 00 02 06 01 00 11 22 33 44 55 66 77 F9 01

Transponder Type: 02_{hex} (Multi-Page)

Page number: 06_{hex}

Status: 01_{hex} (Programming Done)

Page 6 Data: 7766554433221100_{hex}

The data matches that sent so the programming was successful.

6.9 Command 05_{hex} (General Lock Page)

This command is used to lock a page of a Multi-Page TIRIS™ transponder.

6.9.1 Request packet parameter section

Page Number (1 byte).	The page number of the page to lock.

Example:

Lock page 10 of a TIRIS™ transponder.

Request Packet:

02 00 03 05 00 01 00 0A 15 00

6.9.2 Response packet parameter section

Transponder Type (1 Byte).	This byte indicates the transponder type. For a successful lock this should indicate a Multi-Page transponder (02_{HEX}).
Page Number (1 Byte).	This should match the page number of the request.
Status byte (1 Byte).	This byte indicates the status of the page. For successful locking this should be 02_{hex} (Read Locked Page).
Transponder Data (8 Bytes).	This is the data read back from the transponders page after locking.

Example:

Response Packet:

02 00 03 05 00 0B 00 02 0A 02 00 11 22 33 44 55 66 77 FF 01

Transponder Type: 02_{hex} (Multi-Page)

Page number: $0A_{hex}$

Status: 02_{hex} (Read Locked Page)

Page 6 Data: 7766554433221100_{hex}

The data matches that sent so the programming was successful.

+ About TSL

TSL designs and manufactures both standard and custom embedded, snap on and standalone peripherals for handheld computer terminals. Embedded technologies include:

- GPS
- RFID – Low Frequency, High Frequency and UHF
- GPRS/GSM
- IrDA
- Contact Smartcard
- Fingerprint Biometrics
- 1D and 2D Barcode Scanning
- Bluetooth
- 802.11 WiFi
- Magnetic Card Readers
- OCR – B and ePassport

Utilizing class leading Industrial design, TSL develops products from concept through to high volume manufacture for Blue Chip companies around the world. Using the above technologies TSL develops innovative products in a timely and cost effective manner for a broad range of handheld devices.

Telephone: +44 (0)1509 238248
Fax: +44 (0)1509 220020

Postal Address:
Technology Solutions (UK) Limited,
Suite C, Loughborough Technology Centre,
Epinal Way,
Loughborough,
Leicestershire,
LE11 3GE.
United Kingdom.

Email: enquiries@tsl.uk.com

TECHNOLOGY
SOLUTIONS UK LTD

ISO 9001 Registered Firm
Certificate No. GB2001267

INTERNATIONAL ACCREDITATION BOARD
Registration No. 0044/1