



BLUETOOTH AUDIO LINK MANUAL

Version 1.01

Doayee.

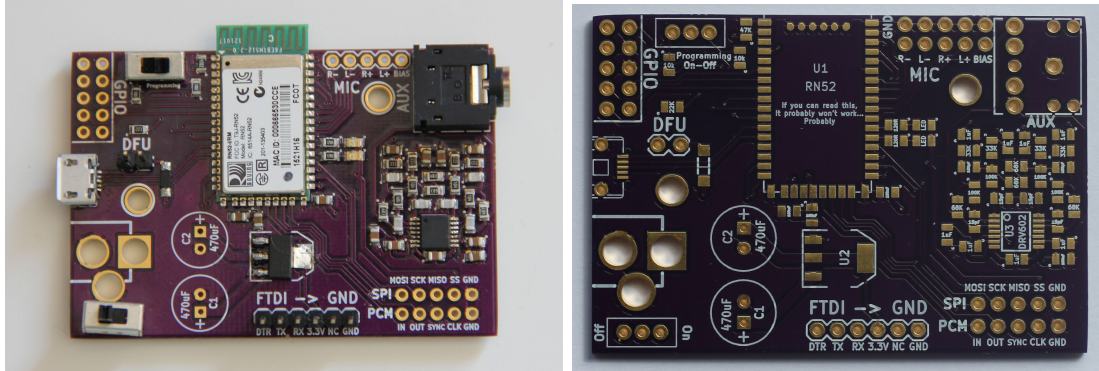
ACKNOWLEDGEMENTS

The BAL was brought to life by the following people, who we can't thank enough.

Adam Boutcher	Eric McNiece	Martin Cook
Alex Barger	Eric Ng	Michael Ferdinands
Andi Kraeuchi	Fated Snowfox	Min Ho Paek
Andrew Camsell	Geoffrey Teekema	Nabeel Chowdhury
Anthony Rippin	Gontarczyk	Nils Rohwer
Ben C. Fisher	Jan Dolinaj	Pete Douma
Bob Sligh	Jean-Claude Ben	Piotr Sieklucki
Bruce Olson	John Marshall	Robert Culver
Callum Brieske	John Vanderlans	Samuel Logan
Chas Anders	Jonny Rothwell	Shaun Biddle
Christopher Brown	Josh Audette	Stefan Kronberger
Christopher G. Stanton	Josh Clark	Timothy Hinchcliffe
Christopher Stanton	Justin Shaw	Tinrit Labs Inc.
David Clark	KADEVICE.com	Tom Bradshaw
David Owens	Krisjanis Linkevics	WLee
Dominic Spill	Lafazar	
Dr.InSide	Mark Mandigers	

For more products by Doayee visit <http://doayee.co.uk>

Bluetooth Audio Link (BAL) Manual



CONTENTS

Page #	Content
3	Introduction
4	Diagram, Basic Operation, Status LEDs
5	Bluetooth Pairing
6-7	Using Command Mode
8	“GET” Commands
9-10	“SET” Commands
11	Setting extended features – special command S%
12	“ACTION” commands
13	Firmware upgrades
14	Advanced info – Bluetooth profiles
15	Use of GPIO
16-17	Construction Information
17	Board Dimensions

BLURB

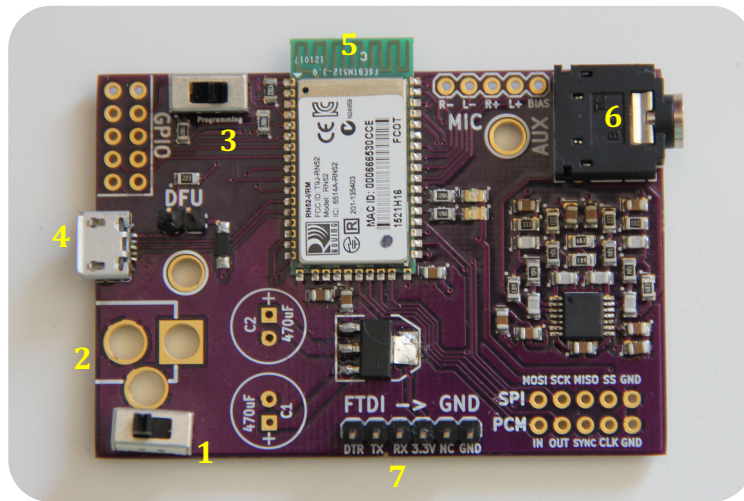
This Bluetooth Audio Link board is a very versatile tool for wireless audio streaming using the RN-52 chip from Microchip. It has a variety of customizable settings, as explained in this guide, and it supports a variety of Bluetooth profiles for the advanced user.

QUICK-START GUIDE FOR REGULAR AUDIO STREAMING

If the module is not pre-built, it *should* be configured with the correct software to play music from the get-go. The extended features (see “SET” commands chapter) are set to hex:2004 by default, which enables discoverability. However if not, investigate the “SET” command S% to find out how to change this.

If module is pre-built (most cases) connect smartphone/tablet/PC and play music. See the next page for more information.

QUICK START GUIDE



The photograph above labels the key features of the board.

- | | |
|-----------------------------------|--------------------------------|
| 1. Power switch | 5. RN52 Bluetooth chip |
| 2. DC Power jack input (optional) | 6. 3.5mm AUX output to speaker |
| 3. Programming Switch | 7. UART Programming Interface |
| 4. USB-Micro Power input | |

BASIC OPERATION

If you had your board pre-built by us, we will have upgraded the firmware of the Bluetooth chip to the latest version, v1.16 and configured the chip to the optimal setup.

This means that the BAL will be discoverable on startup and will also attempt to reconnect to the most recently connected device when powered on. To get started simply plug in a DC source and flick the power switch. The LEDs will illuminate. Once the chip has booted, the LEDs will flash red and green meaning that the module can be found using a Bluetooth device.

Simply navigate to your devices Bluetooth page and look for BAL-XXXX where XXXX is the last four digits of the MAC address printed on the RN52 on the board.

STATUS LEDs

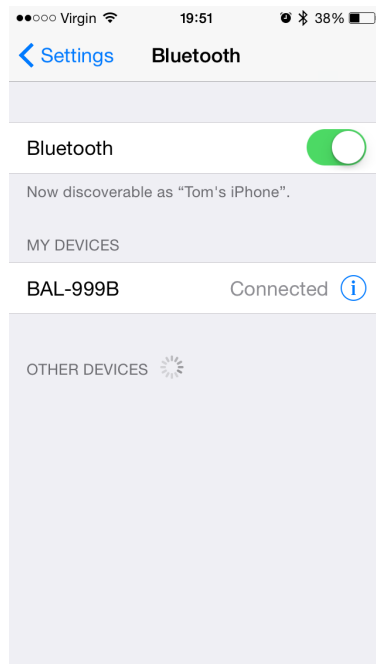
Provided you have connected the LEDs correctly, where LED0 on pin 33 (top) is red and LED1 on pin32 (bottom) is green. LED0/pin33 is the connection closer to the antenna on the RN52.

Red LED	Green LED	Description
Flashing	Flashing	The RN52 module is discoverable
Off	Flashing	The module is connected
Flashing	Off	The module is connectable

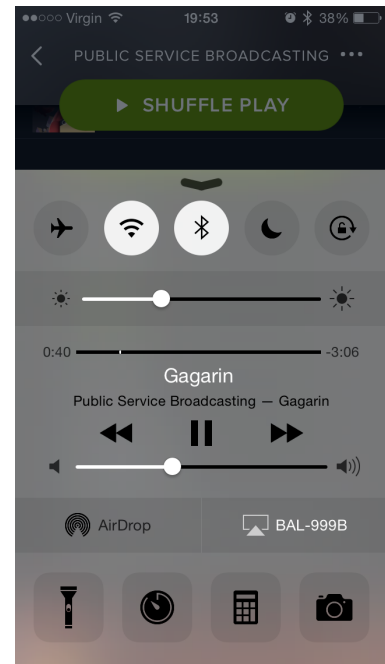
Mobile Device Pairing



1. Tap the Device

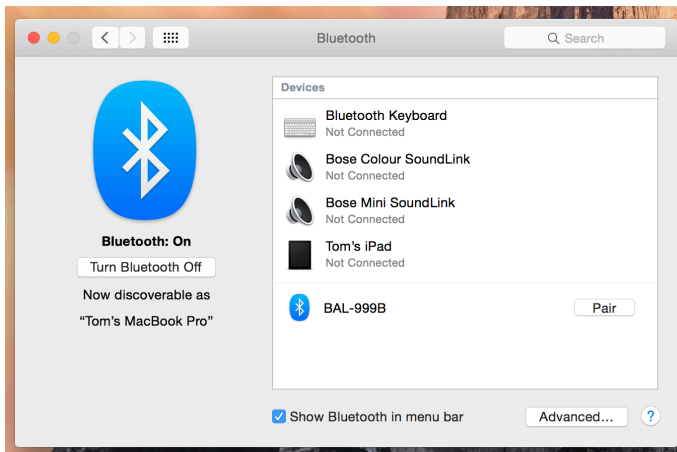


2. Device shows connected



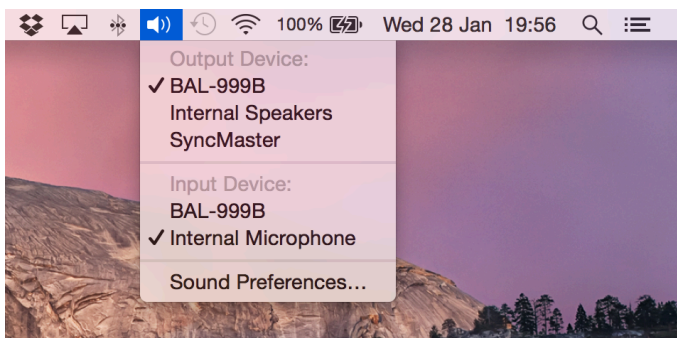
3. Audio is now streaming to BAL

Desktop Pairing



1. Select pair

Of course, the set up process is slightly different for all devices, however the fundamental idea is the same.



2. Use BAL for sound output. Alt-click for this menu on mac

USING COMMAND MODE

The BAL module can have its many settings modified using the UART interface with the RN52 Command mode using a very simple ascii language.

NOTE: This guide contains only commands that we have found the most useful, so as not to inundate you with information, however for the full list of commands read the official RN52 BLUETOOTH AUDIO MODULE COMMAND REFERENCE USER'S GUIDE from Microchip.

Available at <http://ww1.microchip.com/downloads/en/DeviceDoc/50002154A.pdf> or via a quick Google search!

USING ARDUINO LIBRARY

There are two options for communication with BAL, one of which is using a terminal application (skip this paragraph and see below,) or alternatively using a microcontroller interface. For this purpose we have written a library which can be found at: <http://doayee.co.uk/files/RN52LibraryCommandGuide.pdf> or by visiting <http://doayee.co.uk/bal/library>

HOW TO CONNECT TO COMMAND MODE

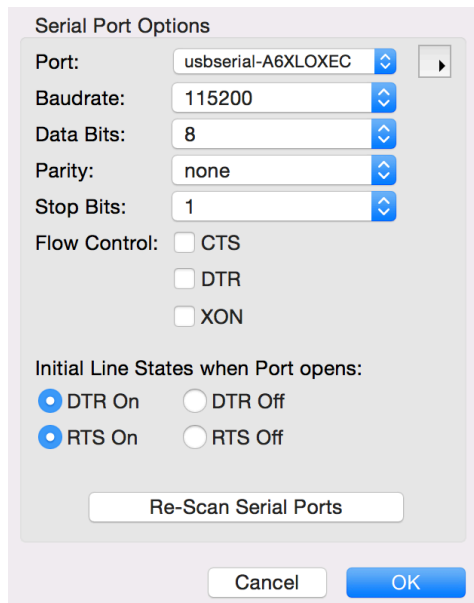
Command mode is accessed using the UART interface. We recommend a SparkFun FTDI adapter.

You will also require a piece of terminal software. We recommend CoolTerm although anything you know how to use will suffice.

Connect your BAL to your computer and find it using the terminal emulator. The default settings for connection are as follows. **The majority of this will already be correct with the exception of the Baud Rate that you are likely to have to set to 38,400 Kbps. NOTE: If you have built the board yourself, the default baud rate is 115,200 Kbps.**

Option	Setting
Bluetooth Mode	Slave
Bluetooth pin code (for legacy pairing mode, can be changed)	1234
Baud rate	38,400 Kbps
Bits	8
Parity	None
Stop bits	1
Flow control	Enabled

Flicking the programming switch on your board (which drives GPIO9 high) will display CMD in your terminal, suggesting the mode is activated, flicking it again will display END, suggesting you have exited command mode.



Serial Port Options

Port: usbserial-A6XLOXEC

Baudrate: 115200

Data Bits: 8

Parity: none

Stop Bits: 1

Flow Control: ☐ CTS ☐ DTR ☐ XON

Initial Line States when Port opens:

☒ DTR On ☐ DTR Off

☒ RTS On ☐ RTS Off

Re-Scan Serial Ports

Cancel OK

To connect in CoolTerm, open the settings panel and find the correct port, most likely a usbserial on a mac or a COM on windows (may vary.)

We recommend setting CoolTerm to Line Mode in the “terminal” tab within the options panel.

You may have to hit “re-scan” at the bottom after plugging in your USB cable. Ensure the baud rate setting is correct then hit OK and press the connect button in the toolbar. Flick the programming switch and see CMD appear in the terminal.

The commands are listed overleaf.

“GET” COMMANDS

Commands for finding out information, obviously!

Setting Echo

If your terminal does not have a local echo/line mode, you can send the + command, all typed characters are echoed to the output. Typing + a second time turns local echo off.

Checking Firmware

The majority of the useful commands to use with the RN52 were only introduced with firmware v1.16. Therefore it is advisable to be running this firmware (latest available as of July 2015.)

Sending the V command will read the firmware version, if you have a version below v1.16 it is highly advisable to upgrade, otherwise the rest of this guide will be fairly useless.

For instructions on how to upgrade firmware see the Firmware Upgrades chapter later in this guide.

Display Active Config

Sending this command displays basic settings as follows.

Active Configuration Legend:

BTA – Bluetooth address of the module
BTAC – Bluetooth address of the connected device
BTName – Friendly Bluetooth name of the module
Authen – Active authentication mode
COD – Class of device
DiscoveryMask – Active discovery mask
ConnectionMask – Active connection mask
PinCode – Assigned pin code for legacy pairing
AudioRoute – Active audio route

Displaying stored settings G<command>

This command displays the stored settings for a set command, where <command> is the second character of a set command (a command prefixed by S as listed in guide)

```
Example: GN // Display the device name
Example: G% // Display the extended feature hex
```


“SET” COMMANDS

The most useful section of the guide, these commands usually take effect after the chip has been rebooted. (Use command R, 1 to reboot.) Commands are not case sensitive. Commands respond with AOK for success or ERR if the command fails.

Setting Device Name **SN,<string>**

This command sets the device name, where <string> is up to 20 alphanumeric characters.

Default value: RN52-XXXX where XXXX represents the last 4 digits of the devices MAC address as printed on the chip. To append XXXX to any name use the command S-,<string>.

Example: SN,MyDevice //sets the name to “MyDevice”

Example: S-,MyDevice //sets the name to “MyDevice-XXXX”

Setting Device PIN **SP,<string>**

This command sets the security pin code, where <string> is up to 20 alphanumeric characters. Each time the device pairs successfully, it saves the Bluetooth address. The device can store up to eight addresses on a first in, first out basis. Using this command also erases all stored pairings. You can use the same value that is already set.

Default: 1234

Example: SP,0123 //sets the pin to “0123”

Set to Factory Defaults **SF,1**

Does what it says on the tin. Reboot for changes to take effect.

When performing a factory reset using a microcontroller, send the SF,1 command, wait for 20ms, send the R, 1 command, and wait for the module to return reboot. Then, drive GPIO9 high to exit Command mode.

Example: SF,1 //reset to factory defaults

Setting Bluetooth Profile **SD,<8-bit hex value>**

This command sets the discovery mask, where <8-bit hex value> represents the profiles enabled for discovery. The Bluetooth profiles are represented by an 8-bit hex value as shown in the table below.

Bit Position	Value	Profile	
0	01	iAP	For more info on these profiles, see the final page of this document
1	02	SPP	
2	04	A2DP	
3	08	HFP	

Default: FF

Example: SD,01 // Set the discovery profile to iAP

Shutdown Timer S^,<decimal value>

This command allows you to get the RN52 module to power off after a specified length of inactivity (where no device is connected via Bluetooth.)

We do not recommend using this as an alternative to flicking the power switch, as other components on the board are still active. The decimal parameters are:

0 – Feature disable, on forever

1-600 seconds of idle time

Default: 0

Example: S^, 60 //power off after 1 minute of inactivity

Default volume gain SS,<8-bit hex value>

This sets the default speaker gain levels which are active at power on using an 8-bit hex value from 00 through 0F

Default: 0A

Example: SS, 0F //set default volume to maximum

System tone volume ST,<8-bit hex value>

If you have RN52 system tones enabled (see “set extended features” command) you can change their default volume using this command, so that they are quieter than the audio output if desired.

The 8-bit hex can be set from 00 through 1F

Default: 16

Example: ST, 10

Set Extended Features

S%,<16-bit hex value>

This is the most complicated to use but probably the most useful command available. The bit mapping for the extended features is:

- Bit 0 – Enable AVRCP buttons (see GPIO on pg. 15 for more info)
- Bit 1 – Enable reconnect on power-on
- Bit 2 – Bluetooth Discoverable on start up
- Bit 3 – Codec indicators PIO7 (AAC) and PIO6 (aptX)
- Bit 4 – Reboot after disconnect (otherwise must power cycle)
- Bit 5 – Mute volume up/down tones
- Bit 6 – Enable voice command button on PIO4
- Bit 7 – Disable system tones
- Bit 8 – Power off after pairing timeout
- Bit 9 – Reset after power off
- Bit 10 – Enable list reconnect after panic
- Bit 11 – Enable latch event indicator PIO2
- Bit 12 – Enable track change event
- Bit 13 – Enable tones playback at fixed volume
- Bit 14 – Enable auto-accept passkey in Keyboard I/O Authentication mode

The default value for this setting is 2004. However, for general audio streaming use it is especially useful to have Bits 1, 2, 7 & 9 activated. **We have found that the ideal configuration is to use 50F6 as your hex value, and we have set this on the modules that we have built, and thus recommend you send the command S%,50F6 to your device if you have built it yourself.** If you require a different set of features then the steps which follow explain how to set further options if you are unsure.

00000000000000 would be the default value in binary, where the value on the right is Bit 0, and working backwards to Bit 14 on the left. So if you wanted Bit 0 and 14 active and others inactive, you would require a binary value of 1000000000000001 and if you wanted only Bit 0 and Bit 13 active, you would use 0100000000000001.

Once you have established the binary value you desire, this must be converted to hex... use the Internet for this (<http://www.binaryhexconverter.com/binary-to-hex-converter>) and then send your desired hex value in replacement of our 50F6 example.

Remember; restart your device after changing these settings to make sure everything works correctly. (either power cycle or send the command R, 1)

Example: S%, 50F6 //sets the advanced features to our recommendation

“ACTION” COMMANDS

These commands are often fairly self-explanatory, however for more detailed explanations please refer to the official RN52 BLUETOOTH AUDIO MODULE COMMAND REFERENCE USER’S GUIDE from Microchip.

Available at <http://ww1.microchip.com/downloads/en/DeviceDoc/50002154A.pdf>

****“I” commands for setting GPIO configurations are documented later on page 8.**

Command	Description
+	Toggle the local echo of RX characters in Command mode
@,<1,0>	Toggle whether the module is Bluetooth discoverable(1)/undiscoverable(0)
#,<1,0>	Accept/reject pairing
\$	Put the module into DFU mode
A,<telephone number>	Initial a voice call to <telephone number>
AD	Retrieve track metadata information
AR	Redial last dialled number
AV+	Increase the volume (AVRCP command)
AV-	Decrease the volume (AVRCP command)
AT+	Play the next track (AVRCP command)
AT-	Play the previous track (AVRCP command)
AP	Pause or start playback (AVRCP command)
B	Reconnect Bluetooth recently paired and connected device.
C	Accept an incoming voice call
E	Terminate an active call or reject an incoming call
F	Release all held calls
I@	Read GPIO configuration **
I@,<hex 16>	Set GPIO configuration **
I&	Reads current GPIO levels for input
I&,<hex 16>	Set GPIO levels for output
J	Accept waiting calls and release active calls
K,<hex 8>	Kill the currently active connection
L	Accept waiting calls and hold active calls
M,<flag>	Toggle the on hold/mute function
N	Add held call
O	Connect two calls and disconnect the subscriber
P	Activate Voice Command
Q	Query the current connection status
R,1	Reboot
U	Reset Paired Device List (PDL)
T	Retrieves caller ID information
X,<0,1>	Transfer call between HF and AG

FIRMWARE UPGRADES

As of v1.16 microchip enabled Device Firmware Upgrade (DFU) over UART, however, we recommend using DFU over USB with the BAL as we have given it a USB port suitable for this function. DFU mode is accessed using the jumper to connect the two DFU pins before power is connected. Both the red and the green LED's will blink simultaneously to let you know DFU mode is active, and then may stop blinking altogether.

To upgrade to firmware v1.16 visit the RN52 page on microchip.com (<http://www.microchip.com/wwwproducts/Devices.aspx?product=RN52#documentation>) and download the **RN52 Bluetooth DFU Utility Installer (Windows)** and the correct DFU image to use with it.

The DFU Utility may be best run in compatibility mode for Windows XP, however if it works for you as is, leave it be.

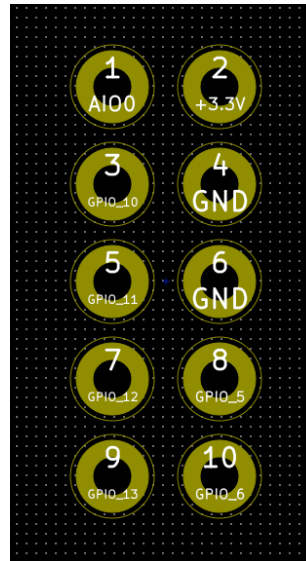
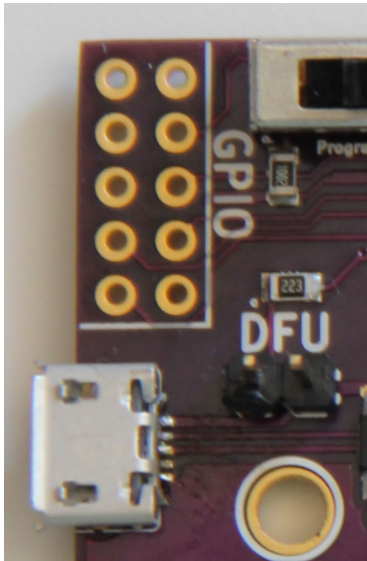
ADVANCED INFO – BLUETOOTH PROFILES

The table below lists the various Bluetooth® profiles supported by the RN-52 onboard the BAL.

Profile	Type	Comments
A2DP	Audio	The advanced audio distribution profile (A2DP) defines how high-quality audio (stereo or mono) can be streamed from one device to another over a Bluetooth® connection.
AVRCP	Audio	The audio-video remote control profile provides a standard interface to control audio/video equipment such as TVs and high-fidelity equipment. This profile is dependent on and used with the A2DP profile.
HFP	Audio	The Hands-Free Profile (HFP) is commonly used in car hands-free kits to communicate with mobile phones in the vehicle.
HSP	Audio	The headset profile provides support for using Bluetooth headsets with mobile phones. This profile is dependent on and used with the HFP profile.
SPP	Data	SPP defines a virtual serial port between two Bluetooth-enabled devices. SPP emulates a bidirectional serial link.
iAP	Data	The RN52 supports Wireless iAP discovery by MFi devices and provides a low level RFCOMM connection to the MFi device. An external host controller with an MFi library is required to perform authentication and iAP protocol implementation.

To explore more advanced “set” commands to toggle which profiles are active on the board read the official RN52 BLUETOOTH AUDIO MODULE COMMAND REFERENCE USER’S GUIDE from Microchip. Available at <http://ww1.microchip.com/downloads/en/DeviceDoc/50002154A.pdf> or via a quick Google search!

ADVANCED INFO – GPIO



The table shows the functions of the GPIO pins, which are mapped to the places on the board as shown to the left.

1	AIO0
2	+3.3V
3	GPIO10
4	GND
5	GPIO11
6	GND
7	GPIO12
8	GPIO5
9	GPIO13
10	GPIO6

These might be useful to you if you are using the product in conjunction with another microcontroller for some purpose. To set the GPIO use the commands referenced here.

Set GPIO direction

I@,<16-bit hex value>

This command configures GPIO direction. A 16-bit mask is used to set GPIO direction: bit on for output, and bit off for input. If no hex is given (“I@” only), the current configuration is returned.

Example: I@,

Set GPIO level

I&,<16-bit hex value>

This command configures GPIO level and input pull-up configuration. Bit on is logical high, bit off logical low. If no hex given, the current levels are returned. The 16-bit <hex16> value is PIO bit map (bit4 = PIO4).

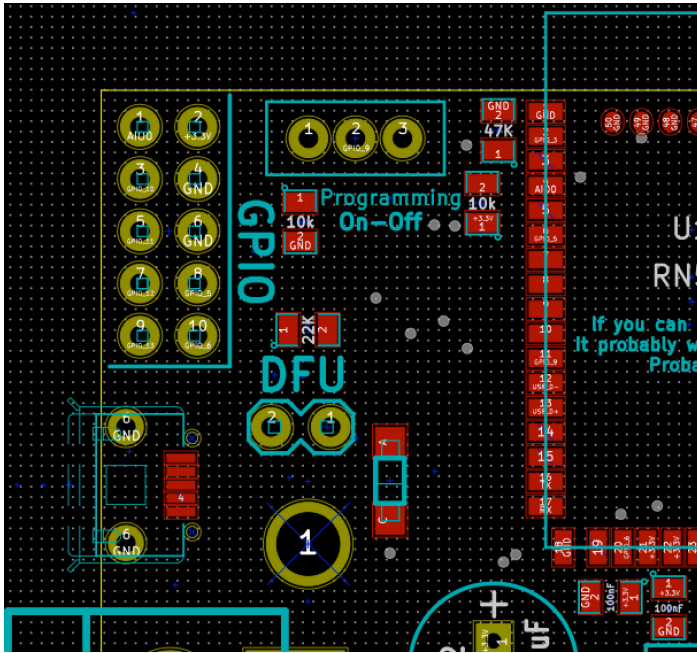
- When PIO is configured as an output, I&,<hex16> drives the level high for PIO specified in <hex16>.
- When PIO is configured as an input, I&,<hex16> configures the PIO to use a pull-up (bit set) or pull-down (bit clear).

Example: I&,

Use of GPIO for AVRCP buttons

If Bit 0 on the Set Extended Features S% command is enabled (see pg. 11), buttons can be attached to the GPIO with the functions shown in this table. Buttons should be connected from the module to ground. Internal pull-up resistors are present in the RN52

GPIO10	Volume Down
GPIO11	Previous Track
GPIO12	Next Track
GPIO13	Play/Pause
GPIO5	Volume Up



We recommend next soldering the RN52 module itself. Be careful not to leave any pins connected to each other, as we cannot be held responsible for damage caused through incorrect soldering.

Finish up by adding the remaining 100nF caps below the RN52 and the other resistors as shown in the diagram. Add the Schottkey Diode to the right of the DFU header. Then add the USB micro port and the barrel capacitors. Please note, the USB is notoriously difficult to solder and will require the

removal of the additional support legs to fit properly.

Finally add the switches (which will most likely require the cutting of additional legs), and optionally the barrel jack, although this is not required for operation of the BAL, as power over USB is adequate.

If you plan to solder the SMT components using a reflow oven, you must first purchase a solder paste stencil. We recommend OSHStencils for this. The Gerber files, including the stencil layer, can be found at <http://doayee.co.uk/bal> under "Open Source". We recommend placing the components in the same order as the soldering instructions provide above, and maintaining a close eye on the soldering process. The temperature and length of time of your heating will depend on the solder paste you are using.

In order to use your BAL with any of our Arduino libraries the baud rate must be changed from its default 115,200 kbps to 38,400 kbps by issuing the command SU,03 (see "SET" commands chapter.)

Board Dimensions

The board dimensions are shown here, for those of you who may wish to produce a case for your BAL.

Please remember the RN52 antenna protrudes from the edge of the PCB.

