



## **Bluetooth Audio Development Pack**

# **Winbond W681360 Codec Board User Guide**

Part Number ACC-005

The information contained in this document is subject to change without notice. EZURiO Ltd makes no warranty of any kind with regard to this material including, but not limited to, the implied warranties of merchant ability and fitness for a particular purpose. EZURiO Ltd shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

© Copyright 2006 EZURiO Limited. All rights reserved.

No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of EZURiO.

Other product or company names used in this publication are for identification purposes only and may be trademarks of their respective owners.

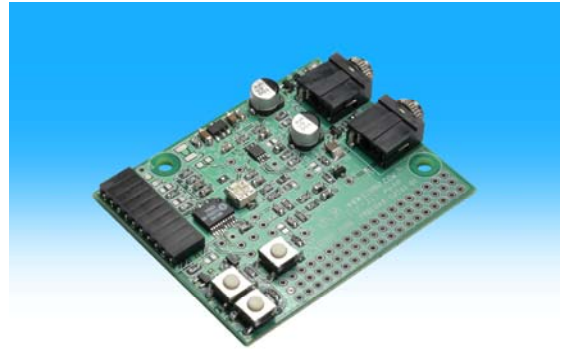
---

# EZURiO

## **Bluetooth® Development Kit**

### Winbond Audio Codec Board

Part Number: ACC-005



## **1. General Description**

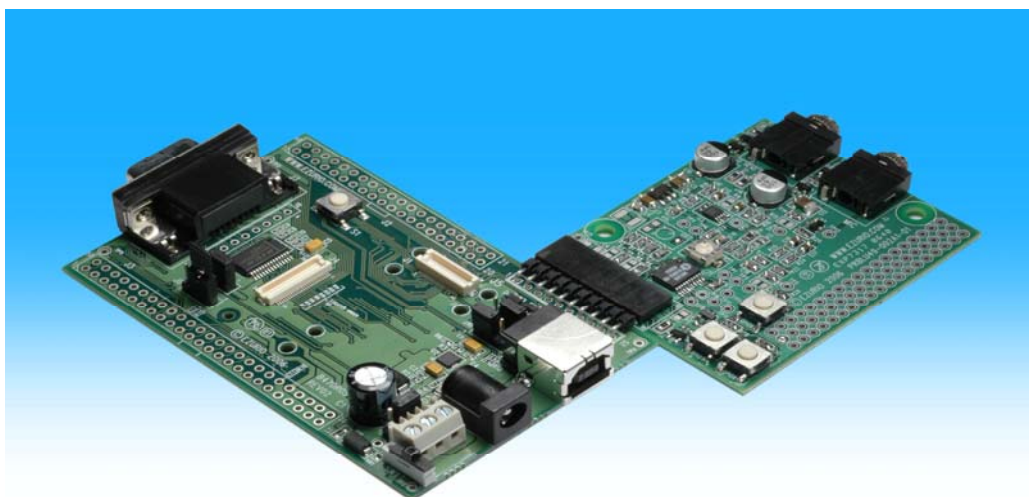
The EZURiO Winbond Codec Evaluation Board plugs into the EZURiO Developers kit and allows you to rapidly test and evaluate *Bluetooth* audio applications using the EZURiO *Bluetooth* Intelligent Serial Module to implement the wireless link.

The ACC-005 evaluation board is based on the Winbond W681360 codec - a 3V, single channel, 13 bit linear voice-band codec, which is pin compatible to the Motorola MC145483. The codec is used to digitise incoming audio from the microphone into PCM data and convert the PCM digital audio output of the *Bluetooth* chip into an analogue signal for the headphones. The codec board has a microphone input and headphone output which are compatible with standard PC headsets.

The W681630 codec has several features such as power down mode and high pass filter disable (to allow frequencies down to DC to be used). The ACC-005 codec evaluation board provides options to allow these features to be tested.

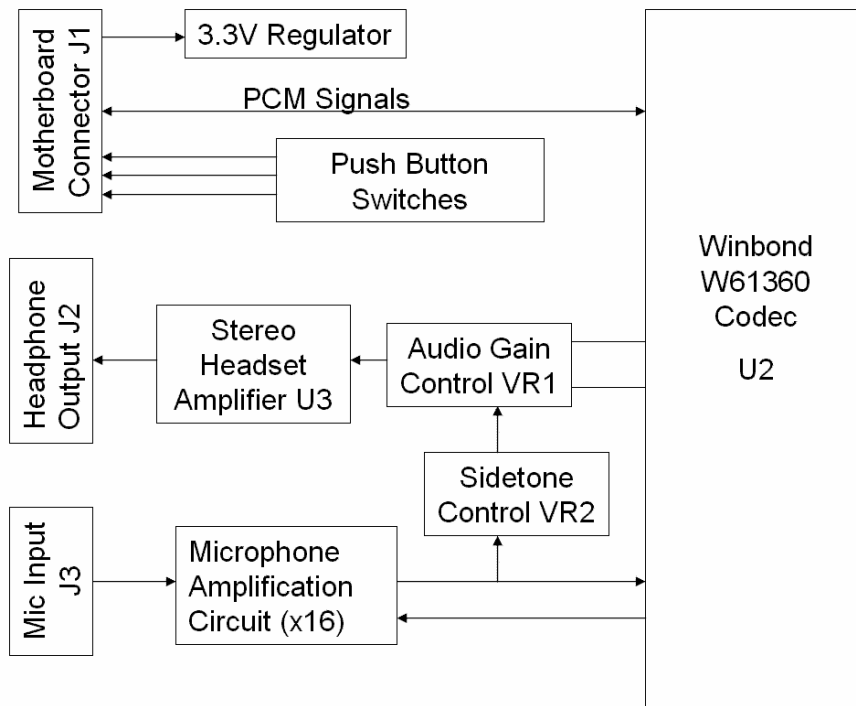
The W681360 incorporates a feature that allows the volume of the codec output to be digitally controlled via 3 bits of the PCM data stream. The BISM II provides an AT command (ATS589) that allows you to control the volume of the codec.

This document provides you with information to prototype and evaluate your own audio application. Once you have tried out your application, you will be able to design your own audio solution based around the Winbond codec and the EZURiO *Bluetooth* Intelligent Serial module.



*Bluetooth* is a trademark owned by *Bluetooth* SIG, Inc., USA and licensed to EZURiO Ltd.

## 2. Overview



The codec board is powered by an on-board 3.3V regulator to reduce noise to a minimum. The PCM control signals for the codec go directly to the *Bluetooth* module on the motherboard via the 10-way connector, as do the 3 push button switches. This allows the switches to be used with an external program that implements the upper portion of headset or Handsfree profile.

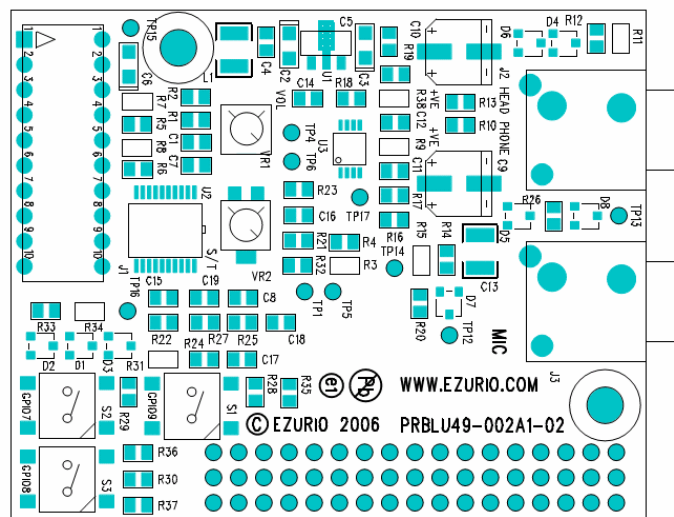
The microphone input, designed to interface to PC compatible headsets, has a fixed gain of 16 set by external components to the codec (the amplifier itself is part of the codec). Part of the microphone signal is mixed into the headphone output signal via VR2. This feature is known as “sidetone” and allows the user to hear their own voice when speaking. It is commonly used in telephony applications to give the user the necessary audio feedback that their ears expect.

The audio output gain is by default fixed at 1. By fitting VR1, the audio gain can be made adjustable.

The 120mW stereo output amplifier U3 ensures that the codec board can drive standard 32Ω stereo headphones while keeping total harmonic distortion down to 0.1%.

### Component Placement

*Note that not all components are fitted – non-fitted components are shown without pads. Refer to Section 7 for details of component fitment and specification.*



### 3. Codec Board Quick Start Guide

#### 3.1 Getting Started

The codec board is supplied with a right angle, 10 way connector that can be used to connect it to the main developers kit. If required, this should be soldered to the main board. Alternatively other connectors or ribbon cables can be used.

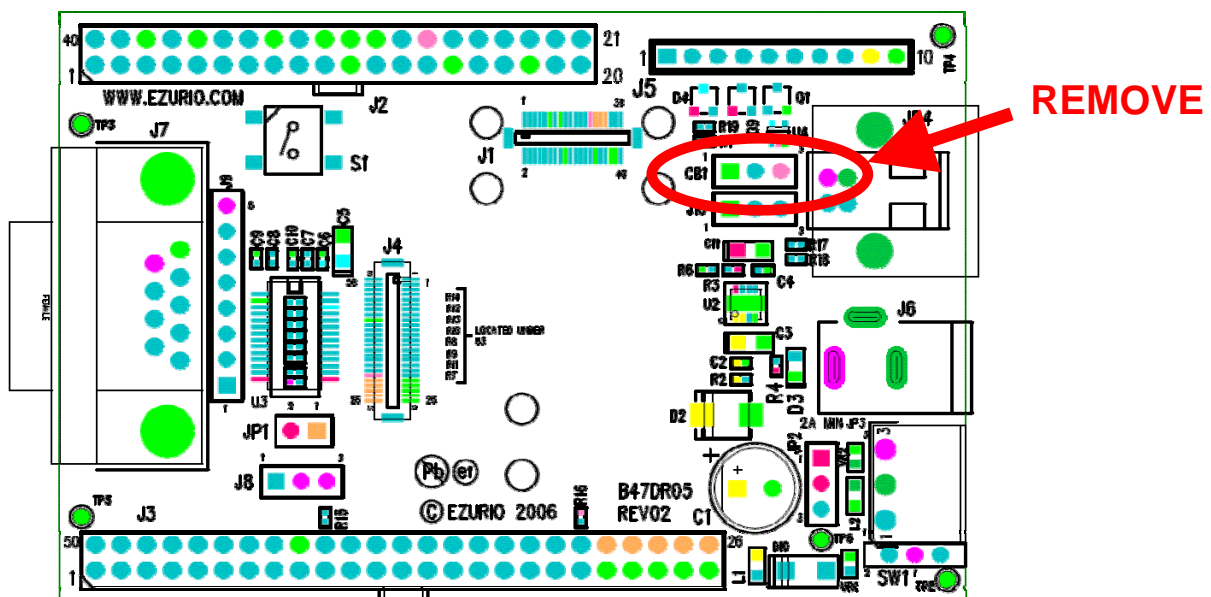
#### 3.2 Equipment Required (not supplied)

- Headsets (with microphone) (Standard PC headsets are fine)
- EZURiO Wireless Developers Kit
- BISM II *Bluetooth* module (Firmware release V9\_20\_22 onwards supports audio volume control)

Normally two sets of development kit are required to test both ends of an audio link. If an application is being developed with an existing endpoint, such as a mobile phone or headset, only one set may be needed.

#### 3.3 Motherboard Jumper Settings

Before using the codec board, there is a jumper setting on the motherboard that needs to be checked. This is CB1, next to the USB adaptor, which **must be removed**. If fitted it will short out the PCM output from the codec and prevent it operating. CB1 is only relevant for the WLAN 802.11 data module.



#### 3.4 Procedure:

- 1) Plug the BISM II into the socket on the Dev Kit, connect to a PC serial port and power up. See the dev kit manual for different power supply options.
- 2) Check that AT commands are working using EZURiO terminal. (Refer to blu2i Quick Start Guide if needed)
- 3) Run the "ATI3" command to find out the firmware release number. If it is less than V9\_20\_22, contact EZURiO to get a firmware upgrade for the BISM II. (Note: older versions of firmware will work, but audio output will be at half the full volume and the ats589=7 command will not be recognised)
- 4) Power down, plug the codec board into the dev kit and power up. Check that AT commands are working.

Configure the Slave unit as follows:

AT&F*	Restore system defaults
ATZ	Reset the unit
ATS512=4	Make connectable and discoverable
ATS0=1	Answer after 1 ring
ATS531=1	Keep AT command mode going after a connection is established
ATS589=7	Set Max. Volume level (requires firmware V9_20_22)
AT&W	Save the above settings
ATZ	Reset the unit.

5) Find out the *Bluetooth* address of the Slave Unit by typing ATi4<return>

6) Configure the Master Unit as follows:

AT&F*	Restore System Defaults
ATZ	Reset the unit
ATS531=1	Keep the AT commands going after a connection is established
ATS589=7	Set volume to maximum
AT&W	Save to flash
ATZ	Reset the unit.
ATD008098nnnnnn	Connect to the slave (substitute your slave's <i>Bluetooth</i> address that you found in step 5 for nnnnnn)
AT+BTA1	Establish an audio link – displays AUDIO ON on both sides. (Alternatively AT+BTA7 can be used and the units will negotiate the best link type.)

An Audio link is now established between the two units.

AT=BTA0 will turn off the audio link (but still leave the units connected).

To change volume use ATS589. ATS589=0 gives minimum, ATS589=7 gives maximum.

## 4. *Bluetooth* SCO Links – A Primer

### 4.1 Normal SCO

*Bluetooth* uses a Synchronous Connection-Orientated link (SCO) for audio. All this means is that for an audio link, the bandwidth needed to maintain the data rates required by the audio link is pre-allocated between the master and slave. This ensures audio data is always transmitted at the required data rate, and takes priority over the transmission of digital data.

The *Bluetooth* specification for SCO is such that there is no re-transmission if data is corrupted or lost. This explains the crackling and popping that occurs when you get to the limits of radio range.

The actual data rate over the air is 64 kbits/sec. There are 1600 timeslots available per second and when a master transmits a SCO packet in one timeslot, the slave replies with its SCO packet in the next. The SCO packet size is fixed at 240 bits (30 bytes). This means when a SCO link is established using the HV3 packet type, two out of every 6 timeslots are used up by the SCO link. This means there is enough bandwidth to have up to three SCO links active between a master and slave at the same time. In this scenario, there are no spare timeslots for other data.

There are 3 main types of SCO packets, HV1, HV2 and HV3 (High Quality Voice). As mentioned earlier, the HV3 packet type has a 1 to 1 mapping between incoming audio data and the data transmitted over the air. There is no error correction possible with HV3.

---

With HV1, each bit is transmitted 3 times and a simple voting algorithm is used at the other end to correct for any bit errors. This means that only 10 bytes of actual audio data can be transmitted in a SCO packet. To maintain the 64 kbits/sec data rate, all 6 timeslots have to be used for the SCO link, leaving no bandwidth available for data.

With HV2, an FEC algorithm is used to correct for 1 bit errors. This increases the data packet size by 50%. This means that only 20 bytes of actual audio data can be transmitted in a SCO packet. To maintain the 64 kbits/sec data rate, 4 out of every 6 timeslots are used for the SCO link.

AT+BTA1 enables HV3

AT+BTA2 enables HV2

AT+BTA4 enables HV1

AT+BTA7 allows the link manager to negotiate which packet type to use, the default is HV1

## 4.2 Enhanced SCO

Enhanced SCO or eSCO was implemented as part of the 1.2 *Bluetooth* Core Specification Release. The main driving factor was to improve audio quality. This has been achieved by:

- 1) including a CRC as part of the audio data packet to allow error detection and a re-transmission request.
- 2) allowing higher data rates by using packets that span more than 1 timeslot
- 3) allowing asymmetric links to allow high quality audio to be streamed in one direction.

eSCO offers significantly better audio quality, but has to be configured at both ends of the link before a unit is enabled to accept incoming connections or enquiries.

To try out eSCO, add the `ATS584=1` command to the commands listed in the quick start section immediately after the `AT&F*` and `ATZ` commands.

Both ends of the link must be configured for eSCO for the audio link to be established. If one end is set to eSCO and the other to SCO, you will get an "AUDIO FAIL" when the `AT+BTA1` command is issued.

The following are the packet types associated with the `AT+BTA` commands for eSCO.

AT+BTA1 – EV3 packet. Up to 30 bytes + CRC. Uses up 1 timeslot

AT+BTA2 – EV4 packet. Up to 120 bytes + CRC + 2/3 FEC. Up to 3 timeslots

AT+BTA4 – EV5 packet. Up to 180 bytes + CRC. Up to 3 timeslots. Currently Unsupported

## 4.3 SCO / eSCO Transport Delays

The following delays have been measured between incoming audio and audio output at the other end of a *Bluetooth* link.

Normal SCO:

AT+BTA1	7.84 ms
AT+BTA2	9.24 ms
AT+BTA4	10.8 ms

Enhanced SCO

AT+BTA1	12.1 ms
AT+BTA2	33.4 ms
AT+BTA4	41.2 ms

As can be seen, the additional error correction of eSCO comes with a transport delay penalty. This is because a buffer is needed to ensure that there is still data to output while waiting for a corrupted data packet to be re-transmitted.

---

For AT+BTA1 and normal SCO, the data is transmitted once every 6 timeslots so the transport delay is expected to be  $6/1600 = 3.75\text{ms}$ . When doing loop-round testing with the codec, i.e. with no transport delay, it was found that from input to output, the codec added  $\sim 1\text{ms}$  of delay at 1kHz and 1.5ms at lower frequencies.

## 4.4 PCM Timing

The codec samples at 8 kHz. The default mode of operation of the codec is 16 bit Receive Gain Adjust Mode. In this mode, in every 8 kHz cycle, 16 bits of data is clocked into the codec. The first 13 bits are PCM audio data, the last 3 bits are volume data. Of the last three bits, 000 equates to maximum volume (ATS589=7), 111 equates to minimum volume (Ats589=0).

At maximum volume, the output signal matches the amplitude of the input signal at the other end of the *Bluetooth* link. It is more appropriate to think of this feature as being an attenuation control.

The clock rate used for sampling is 250kHz ( $4\mu\text{s}$ ). 16 clock cycles takes  $64\mu\text{s}$ . 8kHz equates to  $125\mu\text{s}$ .

The same timing is used for all packet types in both SCO and eSCO modes.

## 5. Frequency Response

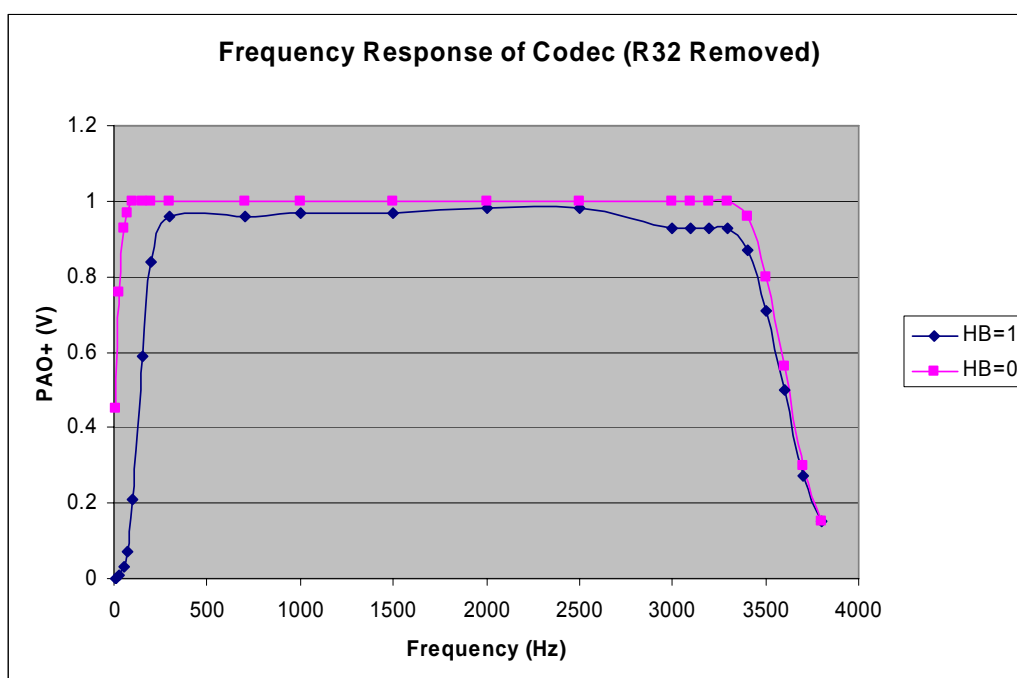
### 5.1 Codec Frequency Response

The codec frequency response can be measured by connecting PCM\_IN from the codec to PCM\_OUT to the codec (PCM\_OUT from J1, the 10 way connector has to be disconnected). A  $1\text{k}\Omega$  pull down resistor is needed on PCM\_OUT to ensure maximum volume setting.

The following graph shows the measured frequency response. For this test, R32, the side-tone resistor was removed to prevent audio feedback.

A 1V peak to peak sine wave was injected into the microphone circuit and its amplitude measured at TP5, A0, the input to the codec. The output from the codec was measured on TP6, PA0+.

The chart below shows the codec frequency response with the High Pass Filter Enable (HB – Pin 16) pin set high and set low.

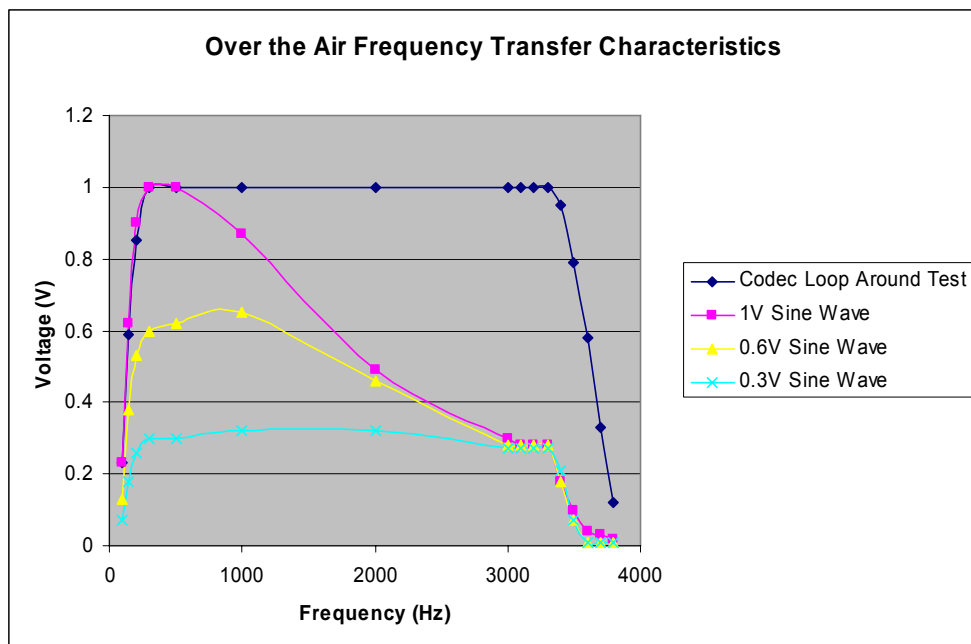


As can be seen from the chart, the codec frequency response is flat between 300 and 3,300 Hz. With the high pass filter on, the 3dB points are at 150Hz and 3,600 Hz respectively. With the high pass filter off, the 3dB point goes down to approximately 15Hz.

## 5.2 Bluetooth Link Frequency Response

The Codec 13bit linear data is coded within the *Bluetooth* chip using CVSD (Continuous Variable Slope Decode) encoding for transport over the *Bluetooth* link. CVSD is essentially a form of Adaptive Differential PCM (ADPCM) and is well suited for voice transmission. It is forgiving of individual bit corruption as each bit only implements an up or a down shift relative to the previous level (corruption of the MSB of a 13 bit sample would create a much larger error term than is possible with ADPCM). A draw back of ADPCM is that it cannot track large delta changes in signal quickly enough. For voice, this does not present a problem.

The chart below shows the frequency response of the *Bluetooth* link at different levels of input sine wave.



As can be seen, the frequency response can only be considered to be flat when the input voltage level is less than a 0.3V peak to peak sine wave.



## 6. Circuit Description

This section describes the individual parts of the circuit and give design information about the components, to allow you to adapt the circuitry of the codec board for your own implementation.

### 6.1 Audio Amplifier

The Winbond codec is capable of driving a  $32\Omega$  load directly if the gain of the output amplifier is reduced by a factor of 4. This is done by Setting R1 to  $39k\Omega$ .

Of the stereo headsets tested, it was found that  $32\Omega$  was a common impedance for each earpiece. For a stereo headset where two speakers are being driven in parallel this would be equivalent to driving a  $16\Omega$  load. This is out of the codec's specification so a small headphone amplifier, U3, has been used on the evaluation board. This is not required if the impedance of the earpiece is equal or greater than  $32\Omega$ .

The large  $100\mu\text{F}$  decoupling capacitors have been used so that the codec could be tested in its "high pass filter mode disabled" configuration. If you do not require a frequency response to go down below 300 Hz, then these capacitors can be reduced to small values. The main design consideration is the impedance should not be significant compared to the impedance of the headphone selected at frequencies of interest.

E.g. if using a  $32\Omega$  headphone and expecting a 3dB point at 300 Hz, then the decoupling capacitor impedance could be  $32\Omega$  at 300Hz i.e.  $10\mu\text{F}$ . This requires a much smaller footprint than the  $100\mu\text{F}$  used in the reference design.

### 6.2 Driving the Headset Directly from the Codec

This will achieve the most cost effective design but care must be taken to ensure that the  $32\Omega$  specification of load is met by selecting an appropriate headset.

Remove R10, R13 and R12. Fit R11, R9, R38 as zero ohm links. Fit  $39k\Omega$  in place of R1 to reduce the gain by 4.

In-house testing showed that with a  $32\Omega$  load and with R1 set to  $39k\Omega$ , that there was some distortion at zero cross-over but that it was not easily perceptible.

Even though the output signal level had been reduced by a factor of 4, on the headsets tested, the volume levels sounded loud enough for most applications. It is important to check this with the target headset for your application.

### 6.3 Microphone Circuit

The microphone circuit is designed for an electret microphone (which is commonly used in PC applications). Typically this would be powered by 5V via a  $2.2k\Omega$  series resistor. In the reference design, it is powered by 3.3V to ensure a clean supply regardless of the power supply used to power the Dev kit. This reduces the sensitivity of the microphone - you should test your application with the microphone and voltage you intend to use in order to determine your component values.

The gain of the microphone is set by R22 and R24, with gain being equal to  $R22/R24$ . The current values are 62K and 3.9K, giving a gain of approximately 16. When changing to a different gain, R27 and R25 should be set to the new values as well. This ensures that the load seen by common mode noise on the microphone is identical and prevents it from being amplified.

R31 is a no fit resistor. It's purpose is to facilitate test modes where a user wants to loop audio output directly back to the audio input to conduct an over the air audio test.

---

## 6.4 Sidetone

When we talk, we hear our own voice, which is part of normal speech perception. If our ears are covered by headphones, we do not hear our voice, which is perceived as abnormal. (Try covering your ears while talking to notice the difference).

To compensate for the loss in feedback to the ear when it is covered with a headphone, most telephony systems inject some of the microphone signal back into the audio output path so that the person perceives their own speech as normal. This feature is commonly referred to as sidetone.

Variable resistor VR2 allows you to control the amount of sidetone that is fed back to the audio output so that the user perceives their speech as normal.

If the headset design does not totally cover the ear, then the sideband circuitry can be omitted.

## 6.5 Power Down

For battery powered audio applications, the power down feature of the codec allows you to turn it off and save power when it is not being used. This feature can be tested by fitting R7 with a 0Ω link and controlling the PUI input of the codec via MPIO\_5.

For AT commands, MPIO\_5 translates to GPIO 7.

The put GPIO 7 into output mode, use “ats610=\$040”

To turn the codec on, use “ats627=1”

To turn the codec off, use “ats627=0”

## 6.6 Alternative PCM\_CLK

Some applications require that the PCM Clock is driven by external circuitry. This requires the PCM Interface provided by the BISM to be put in Slave mode and a clock is supplied by the external circuitry on MPIO\_7.

Contact Ezurio for further details if this is a requirement.

## 6.7 Switches

The switches S1, S2 and S3 have no defined function. They are there to assist you to prototype your audio application. e.g. If your application requires a button to be pressed for the user to answer an incoming connection, you can prototype that function using one of the switches provided.

ATS620 allows you to read the status of the GPIO ports.

No switches pressed:	ATS620? => \$0028
S1 pressed (GPIO 9)	ATS620? => \$0128
S2 pressed (GPIO 7)	ATS620? => \$0068
S3 pressed (GPIO 8)	ATS620? => \$00A8

## 6.8 High Pass Filter Enable

The W681360 can have its High Pass filter enabled or disabled, depending on the state of the HB pin (Pin 16). This is pulled high or low by R3 or R4 (Default). See section 5.1 for more details.

---

## 6.9 GPIO to MPIO Mapping

AT commands use GPIO numbers to represent I/O lines. These GPIO numbers map to physical signals drawn on the schematics as MPIO lines. Some of the GPIO/MPIO lines are used when providing a full RS232 interface.

The following tables gives the mapping between GPIO, MPIO and RS232 signals.

RS232	Hardware
DCD	MPIO_3
RI	MPIO_2
DTR	MPIO_9
DSR	MPIO_8

AT Command	Hardware
GPIO_1	MPIO_0
GPIO_2	MPIO_1
GPIO_3	MPIO_9
GPIO_4	MPIO_10
GPIO_5	MPIO_11
GPIO_6	MPIO_4
GPIO_7	MPIO_5
GPIO_8	MPIO_6
GPIO_9	MPIO_7

Note: For the BISM PA (Class 1 design), MPIO\_0 and MPIO\_1 are used to control the RF switch so are not available to the AT Command Set.

## 7. Bill of Materials

Not all components are fitted, as some provide alternative functionality or implement non-standard options.

Refer to the previous sections and the schematic for information on the component function.

Components marked in *blue* are not fitted.

Reference	Part	Tolerance	Description	Manufacturer	Part No / Footprint
C1,C7	100nF	20%	Ceramic Capacitor		0805
C2,C3,C6	10uF	'+80/-20%	Tantalum Capacitor		TANA
C4,C5,C8	10nF	20%	Ceramic Capacitor		0805
C9,C10	100uF	20%	Electrolytic Capacitor	Panasonic	EEE0JA101SP
C11,C12,C17,C18	2.2uF	'+80/-20%	Ceramic Capacitor		0805
C13	22uF	'+80/-20%	Ceramic Capacitor		1210
C14	100nF	'+80/-20%	Ceramic Capacitor		0805
C15,C19	100pF	20%	Ceramic Capacitor		0805
C16	1.0uF	'+80/-20%	Ceramic Capacitor		0805
D1,D2,D3,D4,D5,D6,D7,D8	BAT54S		Dual Schottky Diode BAT54S	Zetex	BAT54S
J1			10 Way 0.1" R/A PCB Socket	Harwin	M20-7891046
J2,J3			3.5mm 3way Audio Jack Skt	Schurter	4832.232
L1	10uH		Thin Film Inductor		1210
R1,R2,R5,R35,R36,R37	10K	1%	Thick Film Resistor		0805
R3,R7,R8,R9,R11,R34,R38	0R Not Fitted	5%	Thick Film Resistor		0805
R4,R6,R10,R12,R13,R33	0R	5%	Thick Film Resistor		0805
R14,R28,R29,R30	1K	5%	Thick Film Resistor		0805
R15	2K2 Not Fitted	5%	Thick Film Resistor		0805
R16,R17,R18,R19,R24,R25	3.9K	1%	Thick Film Resistor		0805
R26,R20	1.5K	5%	Thick Film Resistor		0805
R23,R21	200K	5%	Thick Film Resistor		0805
R27,R22	62K	1%	Thick Film Resistor		0805
R31	62K Not Fitted	1%	Thick Film Resistor		0805
R32	75K	5%	Thick Film Resistor		0805
S1,S2,S3	OMRON/B3S-1000		Push Button Switch SPNO SMD	Omron	B3S-1000
U1	AME8800AEFT		3.3V Low Drop Out Regulator 300mA	AME	AME8800AEFT
U2	W681360RG		W681360RG CODEC	Winbond	W681360RG
U3	LM4908MM		Dual Headphone Amplifier	Nat. Semi.	LM4909MM
VR1	20K Not Fitted	20%	20K Trimmer	Vishay	TS53YL 20K 20% TR
VR2	50K	20%	50K Trimmer	Vishay	TS53YL 50K 20% TR

## 8. References

- Winbond W681360 Data Sheet – <http://www.winbond.com.tw/PDF/Sheet/W681360.pdf>
- ACC-005 Schematic – ERBLU49-002A1-02

## 9. Disclaimers

EZURIO'S WIRELESS PRODUCTS ARE NOT AUTHORISED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE MANAGING DIRECTOR OF EZURIO LTD.

The definitions used herein are:

a) Life support devices or systems are devices which (1) are intended for surgical implant into the body, or (2) support or sustain life and whose failure to perform when properly used in accordance with the instructions for use provided in the labelling can reasonably be expected to result in a significant injury to the user.

b) A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

EZURiO does not assume responsibility for use of any of the circuitry described, no circuit patent licenses are implied and EZURiO reserves the right at any time to change without notice said circuitry and specifications.

### 9.1 Data Sheet Status

This data sheet contains preliminary data for use with Engineering Samples. Supplementary data will be published at a later date. EZURiO Ltd reserve the right to change the specification without prior notice in order to improve the design and supply the best possible product.

**Please check with EZURiO Ltd for the most recent data before initiating or completing a design. Designers should check the production status of any engineering firmware used during development before it is deployed.**

---