![CML Microcircuits logo]

Features

- **Demonstrates CMX655D functionality**
- **Two digital top-ported MEMS microphones**
- **Easy screw connection for external speaker**
- **Flexibility to use local or off-board power for the PA**
- **Accessible GPIO and test pads to allow further options (e.g. external MEMS)**
- **User modifiable prototype space**
- **Option to add line out phono connector**
- **Simple GUI interface with open-sourced drivers**
- **Conforms to the HAT definition**
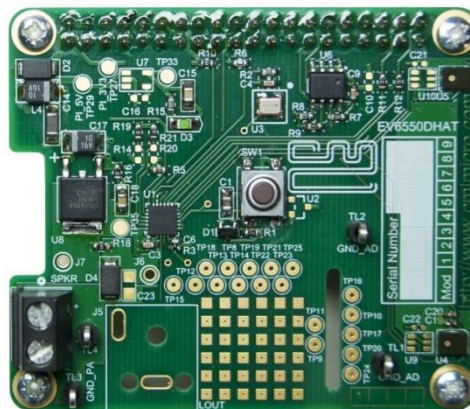
1   Introduction

The EV6550DHAT brings together the Raspberry Pi and the CMX655D Ultra-low Power Voice Codec in a convenient way, enabling easy demonstration of the CMX655D and providing improved low power audio capabilities to the Raspberry Pi. This is achieved using common hardware and open-source software, simplifying its use and simplifying software and hardware development.

2   CMX655D Ultra-low Power Voice Codec

The CML CMX655D Ultra-low Power Voice Codec integrates two matched channels supporting a large variety of digital MEMS (Micro-Electromechanical System) microphones and signal processing capability along with a 1 Watt high efficiency Class D speaker amplifier.

As an inherently low power codec it is ideal for use in low power projects and developments typically based on the Raspberry Pi.
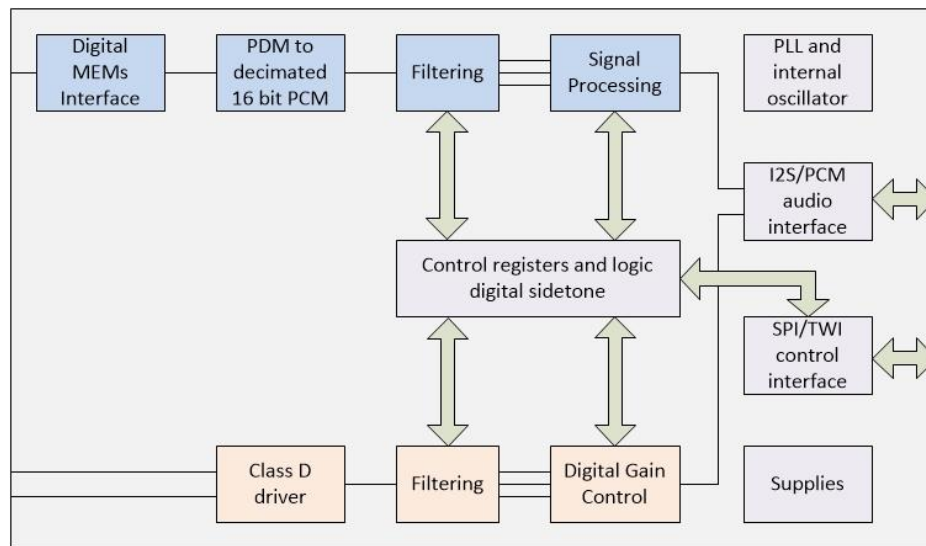
Figure 1 CMX655D Functional Block Diagram

To support evaluation and product developments, the EV6550DHAT is effectively a hardware/software solution that conveniently and inexpensively brings together the CMX655D and Raspberry Pi. The reader is encouraged to read the full CMX655D datasheet before progressing with the EV6550DHAT.

CONTENTS

History

Version	Changes	Date
A	Advance Draft	6th November 2019
1	First public release	11th November 2019

This document is Provisional status and will be subject to future modification. It should not be relied on for final product design.

3   Raspberry Pi Compatibility

The EV6550DHAT has been designed from the offset to offer HAT compatibility and to meet the requirements of the HAT definition. Further details of the format and its requirements can be found on the Raspberry Pi Foundations website: www.raspberrypi.org

4   Power Supplies

EV6550DHAT by default takes power from the Raspberry PI GPIO pins:

- PA$_{VDD}$ Supply: The CMX655D integrated Power Amplifier requires most of the power. A regulator, LM1117DT-3V3, is fitted to the HAT PCB and sources 5V from the GPIO connector supply, routing 3.3V to the PA$_{VDD}$ pin via an on-board LDO regulator. There are options to use external supply via links and pads if required.

- AD$_{VDD}$_3V3 Supply: The remaining CMX655D power by default is supplied via a low current regulator, LP2985IM5X-3.3. An alternative 3V3 supply is available from the Raspberry PI GPIO 3.3V pins via a decoupling inductor to reduce the digital noise from the Raspberry Pi supply. The regulator and capacitor should be removed and a suitable inductor mounted to use this option. The resulting 3.3V is routed to the CMX655D low powered circuits via the CMX655D VDD_AD1 and VDD_AD2 pads. There are options to use an external supply via links and pads.
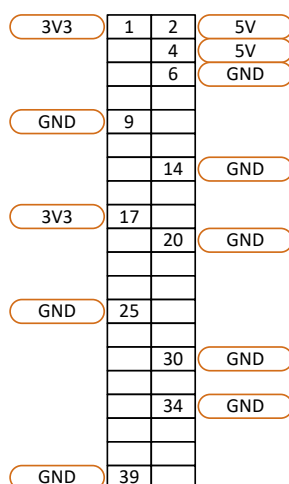
3V3	1	2	5V
		4	5V
		6	GND
GND	9		
		14	GND
3V3	17		
		20	GND
GND	25		
		30	GND
		34	GND
GND	39		

Figure 2 40-way GPIO Connector Power Pins

Note: The 3.3V Raspberry Pi pins are limited to a maximum of 50mA, which is sufficient for operation of the CMX655D and associated components. The 5V supply has greater current sourcing capability and as such by default is used to supply the PA via a small regulator. However, the amount of current that can be sourced from the 5V pin is very dependent of what other USB peripheral devices are mounted as well as processor power consumption. For a 4 Ohm speaker load typically 400mA can be drawn from the 5V rail at maximum signal level. Should there be an issue it is recommended that an external 3.3V supply is used to power the PA, there are links and pads available for this change.

Note: If an external PAVdd supply is to be used instead of the PAVdd regulator, the PAVdd regulator must be disconnected (0Ω link removed) from the supply.

5   Handling Precautions

In common with other CML Evaluation Kits, the following practices will help ensure proper operation.

Static Protection

This product uses low-power CMOS circuits that can be damaged by electrostatic discharge. Partially-damaged circuits can function erroneously, leading to misleading results. Observe ESD precautions at all times when handling this product.

6   Quick Start

6.1   Mounting The EV6550DHAT To The Raspberry Pi

The EV6550DHAT follows the Raspberry Pi Foundations common HAT format so the card will plug into a Raspberry Pi with no modifications. It is compatible with any Raspberry Pi with a 40-pin extended GPIO connector.

With the Raspberry Pi in an unpowered state, the EV6550DHAT can be mounted by pushing the board onto the 40-way GPIO connector. Additionally 4 metal standoffs and 4 screws are supplied with the EV6550DHAT that may be used to support the board via 4 mounting holes positioned to fit the Raspberry Pi board's holes.

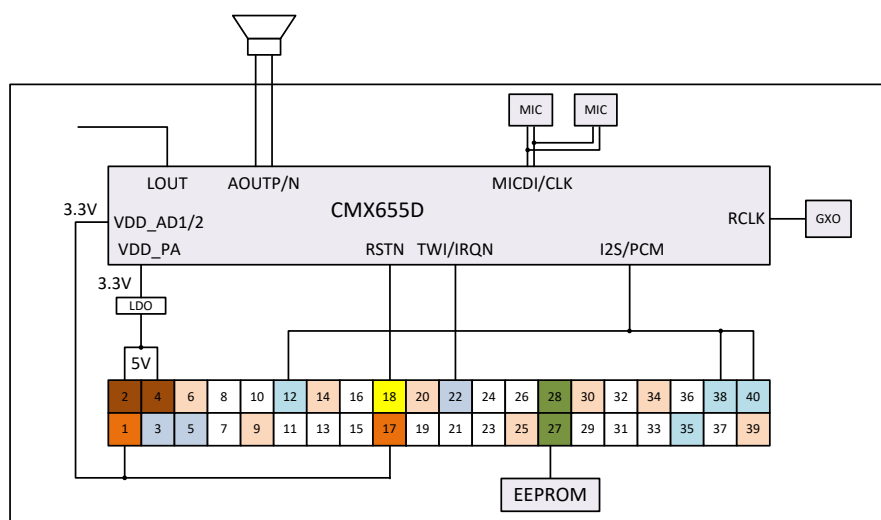The EV6550DHAT uses the following GPIO to communicate with the Raspberry Pi.

Figure 3 EV6550D 40-way GPIO Signal Path Routing Diagram

6.2   Software Installation

EV6550DHAT hardware and software is supplied as a single package, with HAT device tree data pre-loaded onto the EV6550DHAT's on-board EEPROM. To enable complete and full operation the following applications should be installed. It is assumed that the latest Raspbian version "Buster" is resident in the host Raspberry Pi:

GTK3+	A cross platform graphical user interface tool kit and API	www.gtk.org
GEANIE	A programming text editor supplied with Raspbian	www.geany.org
GLADE	A Rapid application development tool (RAD) for GTK3+	glade.gnome.org

These applications can be considered optional; there are other alternative open-source software tools that are compatible and can be used if modification of the EV6550DHAT software is required.

While it is possible to update the EEPROM HAT via third party tools the current data file includes a unique board specific identifying UUID value that should be retained wherever possible. This serial number is necessary to retain manufacturing traceability of the EV6550DHAT.

Note that the EEPROM is write protected via pull-up resistor R7. Test point TP26 should be pulled low to remove this protection if necessary.

6.3   GUI Description and Board Operation

To execute the EV6550DHAT GUI from the Raspbian desktop, locate and click the RunGUI.sh script, the GUI should launch. Alternatively the EV6550DHAT can be launched from the terminal as root:

    sudo ./HATGui

The user should ensure that necessary permissions are in place to allow the shell scripts to execute. In some instances permissions may not be retained when files are moved and so need to be set appropriately for the shell scripts and executable files.
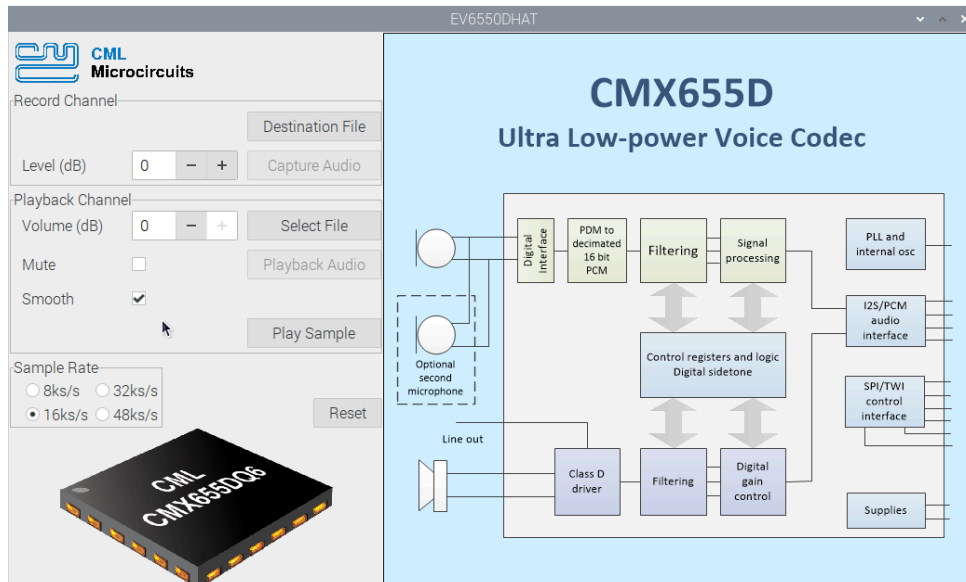

Figure 4 EV6550DHAT GUI

The GUI provides access to pre-recorded sample files and the capability to record or playback user .wav files at 8, 16, 32 or 48ksps. Volume and Level adjustment is possible via dedicated "spinners" and the facility to mute or smooth the playback volume is also available.

Record Channel

This function allows audio to be captured from both MEMS microphone and saved locally as a .wav file. They are written left channel then right channel, where MEMS U4 is the left audio channel and MEMS U5 is the right audio channel.

The "Destination File" Button opens a window to the location where the resulting .wav file will be stored. The file is called "Capture" by default but this may be modified by the user. It is suggested that the chosen sample rate is annotated into the file name for easier identification and retrieval.

Once chosen the "Capture Audio" button will be displayed and an audio recording is possible. Input audio gain can be adjusted using the level spinner from -12dB to +3dB in 1dB steps. While the capture process is active, remaining GUI functionality is disabled.
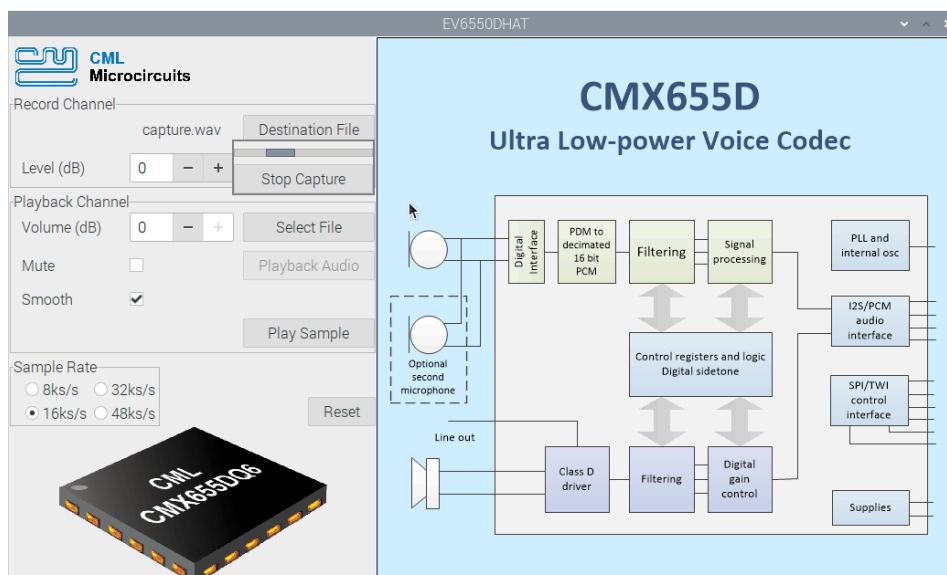
Figure 5 EV6550DHAT GUI - Recording Audio

Playback Channel

This function allows pre-recorded audio to be played through the speaker output in mono via the CMX655D Class D amplifier.

The "Playback Audio" button becomes available when a compatible wav file has been selected via the "Select File" button.

The sound file selected should be S16_LE, Stereo format at the applicable rate of 8000, 16000, 32000 or 48000bps. Once selected, a panel is launched when retrieving and playing the audio. A progress bar and gain indicator are displayed. Audio is taken from the left channel or MEMS U4. Overall playback level may be adjusted from -90dB to 0dB in 1dB steps. Mute and Smooth tick boxes are also available, where smooth eliminates any audible distortion while volume is being changed rapidly and mute sets the playback level to maximum attenuation (-90dB).
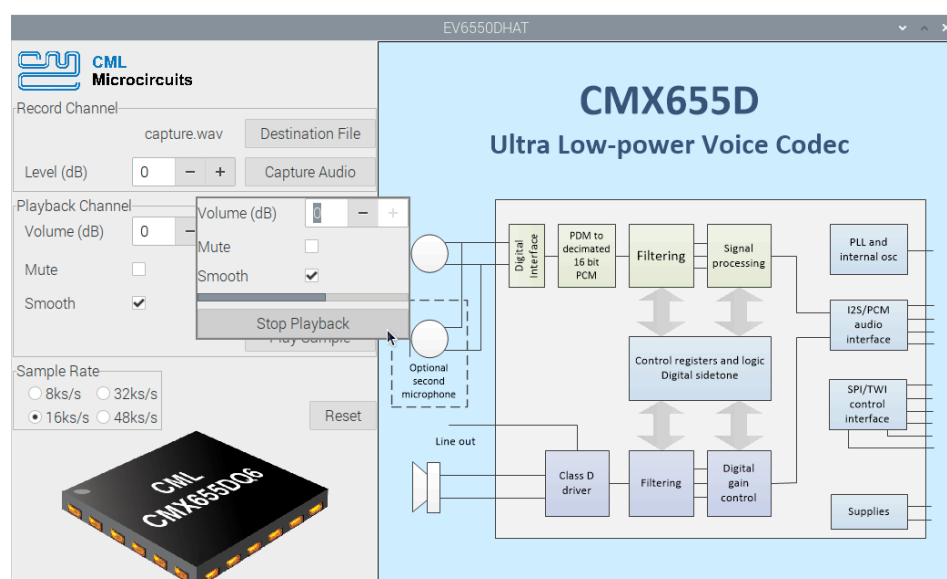

Figure 6 EV6550DHAT GUI - Playing Audio

The "Play Sample" button is included as a convenient means of comparing 4 pre-recorded samples at 8, 16, 32 or 48ksps for quick comparison of sample rate vs audio quality.

Thermal and Current Protection is provided to avoid the use of external support circuitry. The CMX655D includes integrated thermal and over-current detection circuits which indicate to the GUI that the integrated Class-D amplifier is being driven too hard. The EV6550DHAT GUI can be used to re-enable the CMX655D if such a state occurs or alternatively the Reset button may be pressed.

There are two ways to reset the EV6550DHAT: via the reset button in the EV6550DHAT GUI or by depressing the board-mounted reset button. The reset button is included to support user development with the EV6550DHAT in applications where a hardware reset is required. If the reset button on the EV6550DHAT is pressed whilst using the supplied firmware the CMX655D will be reset but not the GUI software (which may behave unexpectedly, e.g. invalid IRQN). It is therefore advisable to perform a reset via the GUI software and not the on-board reset button.

7   Supplied Source Code

The supplied EV6550DHAT software consists of the files shown in Figure 7. A description of each file type is given in Table 1.
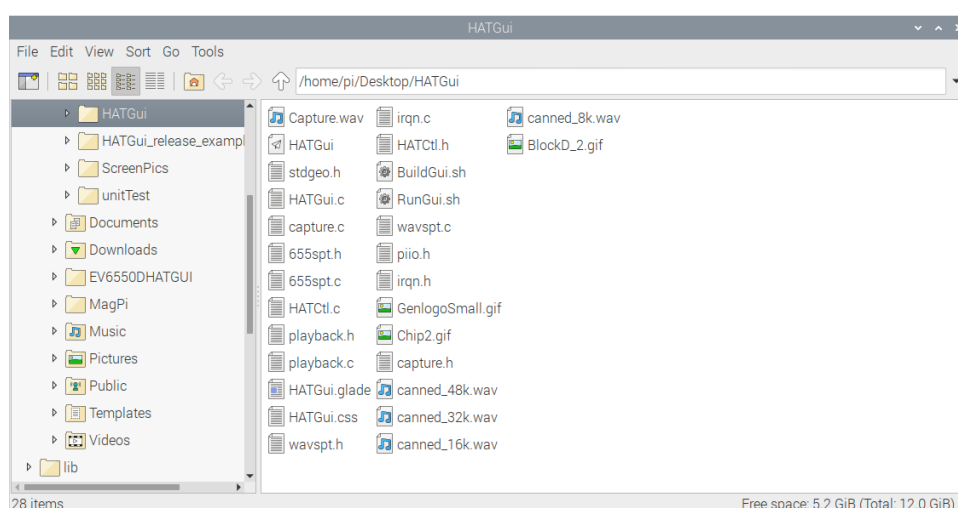

Figure 7 EV6550DHAT Software Files

Table 1 EV6550DHAT Software Files

Source ".c" and Header Files ".h"

655spt	CMX655D control and communication commands via the I2C and I2S interfaces
Capture	Voice capture and record functions via alsa arecord, uses dedicated dialog window and uses "arecord" to facilitate audio capture.
HATCtrl	Defines GUI widgets and user interaction.
HATGUI	GUI style and interoperates with GLADE
irqn	Basic interrupt servicing of the CMX655D via a status register polling mechanism.
playback	Voice access and playback via alsa aplay using dedicated dialog window
wavspt	Supports the use of .WAV files used for the supplied voice samples and user-generated sound file.
piio	A header file for PI internal peripherals, specifically GPIO, CLK, I2S and I2C.
stdgeo	Any additional functions will be added here

Shell files ".sh"

BuildGui	Shell script that builds the GUI. This shell script directs the GNU Compiler Collection (GCC) to the source files, Libgpiod API and the GTK+ header files and libraries.
RunGui	Shell script that runs the GUI as root. A convenient solution to allow the GUI to run as root from the desktop not via a terminal.

GLADE files ".Glade" and ".css"

HATGui	.GLADE project file used by HATGUI.c and HATGUI.h files. This simplifies creation and modification of the GUI interface.
HATGui	.css GUI file for defining any GUI formatting or styles applied to the GUI interface.

Images ".gif"

BlockD_2	CMX655D Block Diagram used by the GUI
Chip2	Chip illustration used by the GUI
GenlogoSmall	CML logo used by the GUI

Sound sample files ".wav"

Canned_xxx	Four pre-recorded 16 bit voice samples at 8k, 16k, 32k and 48ksps

API Instruction File

EV6550DHAT_API.doc	A reference guide for the EV655DHAT API. This document lists and describes commands that are implemented in the 655spt.c source file, some of which are used to enable EV6550DHAT GUI functionality.

8   GUI 'C' Source Files and Modification

As the source files for the GUI are provided, it is possible to recompile the executable within a suitable text environment. In this example Geany is used (which is supplied as part of the Raspbian OS).

Note: CML does not accept responsibility for any changes made to the source code should the EV6550DHAT fail to operate as expected or be damaged as a result of any improper modifications.

Before proceeding it is important that access permissions are set to "all users" for all scripts files ".sh" and the executable HatGui if the user is not already the "super user". See online Raspbian documentation for hierarchical structure and access.

From within Geany, the supplied build files can be opened and the BuildGui.sh script selected. The basic build commands can be selected from the menu. Figure 8 shows a simple "MakeMe" setup. Here "BuildGui.sh" is used to link and compile the C source files and the executable run via "RunGui.sh" from within the Geany application.
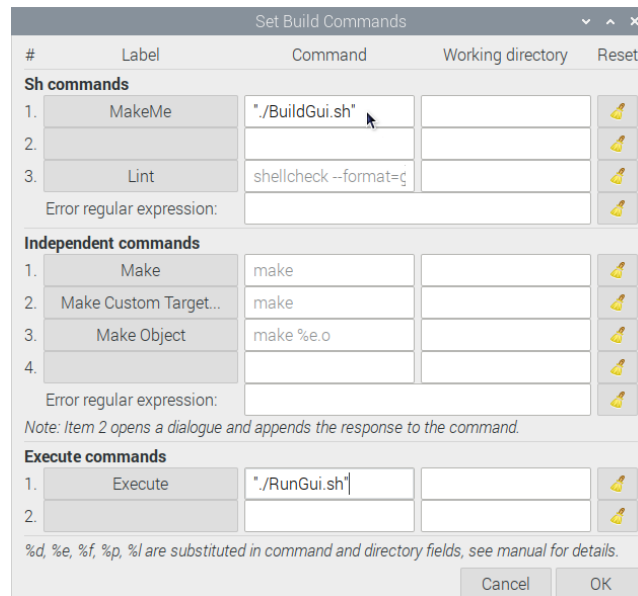
Figure 8 Geany Build Screen

The GUI API calls used to communicate and control CMX655D on the EV6550DHAT are defined in the 655spt.c source file. The EV6550DHAT GUI-specific API commands are further described in the supplied EV6550DHAT_API.doc file. The 655spt.c and 655spt.h files include a number of API calls in addition to those used by the GUI. This facilitates modifications to the GUI or user application to suit the user application or setup. For even more control of the CMX655D via its control registers it is possible to write to and read from the I2C interface directly by using CMX655WriteByte(), CMX655Write(), CMX655ReadByte() and CMX655Read() commands.

9   Troubleshooting

Error Observed	Possible Cause	Solution
"HATgui" not recognised	OS not the latest version	Update the Raspberry Pi OS to "Buster" or higher.
	The necessary files have not been installed.	Ensure that GTK3 has been installed before using the EV6550DHAT
"HATgui" does not run	User does not have "superuser" privileges	Either run as a superuser in the Raspberry Pi terminal or run the shell script HATgui.sh from the desktop.
		Ensure that all executable file permissions are set to "everyone". Specifically RunGui.sh and HatGui. If necessary the same is true for BuildGui.sh.
Playback window closes when attempting to open a .wav file.	Incompatible .wav file format	Check that the message "Data format not supported" is reported in the terminal. If it is, ensure that .wav file is compatible with the S16_LE, Stereo.wav file format .
Dialog message "Sample rate not supported" when selecting a sound file.	.wav sample is not supported by the "HATgui"	Playback sample rates are 8, 16, 32 and 48kHz.
Audio sounds "squeaky" or "slow"	Wrong sample rate applied to the selected .wav	Ensure that the appropriate sample rate is applied for the selected .wav. It is helpful to include sample rate in the file name.
Sound from speaker very low/noisy	Speaker being used is the wrong impedance	Use a 4Ω to 8Ω speaker

The GUI prints additional debug information to the terminal when running the executable via a terminal. See Quick Start section 6.3 for information on starting in terminal.

10   Additional Hardware Information

10.1   Hardware Test Points

Alongside the 6x6 prototype area and across the PCB are test points to enable measurement and adaption where required. These are listed in Table 2.

Table 2 Test Points

Test Point Ref.	Default Measurement	Description/Default State
TP1	RSTN	3.3V
TP2	RCLK	24.576MHz
TP4	VDD_A	GND
TP5	BIAS	$AD_{VDD}/2$
TP8	PI_GPIO4_GCLK	Pull up to 3.3V (GPCLK)
TP9	PI_GPIO5	Pull up to 3.3V
TP10	PI_GPIO6	Pull up to 3.3V
TP11	PI_GPIO7_SPI_CE1_N	SPI Pins
TP12	PI_GPIO8_SPI_CE0_N	
TP13	PI_GPIO9_SPI_MISO	
TP14	PI_GPIO10_SPI_MOSI	
TP15	PI_GPIO11_SPI_SCLK	
TP16	PI_GPIO12	Pull down to Gnd
TP17	PI_GPIO13	Pull down to Gnd
TP18	PI_GPIO14_TXD0	I2C
TP19	PI_GPIO15_RXD0	
TP20	PI_GPIO16	Pull down to Gnd
TP21	PI_GPIO17	Pull down to Gnd
TP22	PI_GPIO22	Pull down to Gnd
TP23	PI_GPIO23	Pull down to Gnd
TP24	PI_GPIO26	Pull down to Gnd
TP25	PI_GPIO27	Pull down to Gnd
TP26	VDD_AD	3.3V
TP27	PI_3V3	3.3V
TP28	PI_3V3	3.3V
TP29	PI_5V	5V
TP30	PI_5V	5V
TP32	ADVDD_3V	3.3V
TP33	ADVDD_3V	3.3V
TP34	PAVDD	3.3V
TP35	PAVDD	3.3V

The EV6550DHAT PCB consists of four layers of FR4 with 2oz copper top and bottom and 1oz copper for layers two and three, with nickel gold finish. EV6550DHAT schematics and gerber layout files are available from the CML website. www.cmlmicro.com. The EV6550DHAT is supplied without an on-board phono connector, should one be needed space and pads are present to allow a wide range of RCA connectors, a typical manufacture such as Lumberg supplies suitable connectors.

CML Microcircuits	United Kingdom	p: +44 (0) 1621 875500	e: sales@cmlmicro.com techsupport@cmlmicro.com
	Singapore	p: +65 62888129	e: sg.sales@cmlmicro.com sg.techsupport@cmlmicro.com
	United States	p: +1 336 744 5050 800 638 5577	e: us.sales@cmlmicro.com us.techsupport@cmlmicro.com

www.cmlmicro.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[CML Microcircuits](#):
  [EV6550DHAT](#)