MagicQuartz 1.0x “Phoenix” Documentation

MATE LABS Driving Your Passion.

About this Version's Bird

The Phoenix is a mythological bird that is said to have the ability to regenerate itself from its own ashes. The Phoenix is typically described as a large, colorful bird with golden or red feathers, and is said to have a lifespan of several hundred years. The Phoenix is often associated with themes of renewal, immortality, and resurrection, and has been featured in various works of literature and art throughout history. It remains a popular symbol today, often used to represent personal transformation and growth.

The Phoenix was selected for this release because the development of MagicQuartz stalled in 2016 due to time constraints. However, the development resumed in 2021 and was completed in 2023. One could say that MagicQuartz rose from its own ashes, much like the Phoenix.

Copyright 2024 Sebastian Mate

This document is subject to change without further notice.
Document revision date: February 10, 2024

AVR is a registered trademark of Microchip Technology Inc. Arduino is a registered trademark of Arduino SRL. All other trademarks, logos and brand names are the property of their respective owners. All company, product and service names used in this document are for identification purposes only.

Contents

Part I.

Software Guide

1. Introduction

Welcome to Precise Turntable Speed!

MagicQuartz is a microcontroller firmware for the accurate speed control of HiFi record players with AC motor. The highly configurable software can be used for control and signal generation in true sine wave inverters, enabling the implementation of precision turntable speedboxes or motor power supplies.

MagicQuartz aims to perfect the playback speed of vinyl records. It not only offers complete decoupling of a turntable from mains frequency fluctuations, but also electronic switching and adjustment of the turntable's speed. It is even possible to have the speed continuously monitored and automatically readjusted. MagicQuartz is also a valuable tool for the reproduction of shellac records, as non-standard speeds can be easily set.

MagicQuartz has been developed for the 8-bit AVR microcontroller family from Microchip (formerly Atmel). In the context of rapid prototyping, the firmware can also be run directly on Arduino hardware, such as the *Arduino Mega 2560 Rev3*. To interact with the firmware, an LCD keypad shield or a serial connection via the Arduino's USB interface can be used.

MagicQuartz is aimed at professional hardware developers from the absolute high-end audio sector, as well as private audio enthusiasts with appropriate knowledge in electronics. A free demo version is available for evaluation and hardware development purposes.

What initially started as a crazy idea soon turned into the ambitious goal of developing the world's best turntable speed control software. The project now looks back on over a decade of passionate research and development. I hope that you will enjoy MagicQuartz!

Sebastian Mate (Erlangen, Germany)

2. Features

MagicQuartz is a feature-rich software. This section provides an overview of its features. Note that some features may not work with all turntables – please see section 3.2 on page 10 for details.

Smart Turntable Speed Control

- □ **Speedbox Function:** Electronic speed switching between the standard record speeds (16, 33, 45, and 78 RPM) without having to change the speed setting on the record player. Smooth AC frequency ramping. *See section 5.1.2 on page 35 for details.*
- □ **Live Quartz Mode:** MagicQuartz continuously monitors the speed of the turntable and corrects the generated AC frequency if necessary so that the turntable maintains the correct speed. *See section 5.3 on page 40 for details.*
- □ **High Timing Resolution:** The system clock, speed measurement, speed computations, and signal generation operate on a high temporal resolution of 100 µs.
- □ **Intelligent Control:** When larger speed changes occur, the regulation automatically stops, for example, while

the start/stop functions of an automatic turntable are in operation. In the event of a fault, such as a motor failure, broken belt or idler wheel, the speed control also stops automatically.

□ **High Customizability:** Adjust the regulation velocity, set limits for the allowed frequency change, and fine-tune many other parameters to suit your needs and preferences.

□ **Voltage Reduction:** Optionally, the generated AC voltage can be reduced automatically in “Live Quartz Mode”, which may lead to a significant reduction of motor noise (i.e., hum caused by vibration) in some turntable motors.

□ **Transpose Function:** Musicians can easily change the pitch of the playback in semitones.

True Sine Signal Generation

□ **Multiple Wave Forms:** MagicQuartz supports the generation of single-phase or dual-phase sine wave signals, the latter with variable phase shift. To realize more complex inverter designs, the single-phase sine wave can also be output separately on two channels in the form of a negative and positive half-wave. Optional software pop suppression for less complex inverter designs. Configurable, smooth up and down ramping of the AC voltage.

□ **HiFi-Compatible:** The digital to analog signal conversion is implemented using pulse width modulation

(PWM) at a carrier frequency of 31.250 kHz, which is beyond the range of human hearing.

- □ **High-Resolution:** The signal is based on 8-bit wavetables with 128 samples for each full sine period, scaled using floating point arithmetic and reproduced at a variable sampling rate (variability in true 0.0001 Hz steps). The latter is achieved by a clever programming of the microcontroller's hardware timers.

See section 8 on page 83 for details on the signal generation.

Advanced Power Management (APM)

- □ **Current Monitoring:** Continuous measurement of the inverter's current at a sample rate of up to 5 Hz.
- □ **Voltage Correction:** Detection of the motor switching status based on the inverter load, followed by a correction of the generated voltage.
- □ **Error Handling:** Detection and handling of inverter shutdown and overload.

See section 9 on page 91 for details on the APM.

Other Features

- □ **Motion Sensor:** Automatic motion sensor calibration that also guides the user when positioning the sensor on the turntable.

□ **Keep Track:** Uptime counters that also count the on-time of the record player, which can be used to keep track of stylus wear.

□ **Little Distraction:** When turned on, MagicQuartz automatically continues in the mode it was last in, allowing the user to focus on the music and not on MagicQuartz.

□ **For Longevity:** Advanced flash memory caching and wear leveling to prolong the life of the microcontroller.

□ **For Safety & Environment:** Configurable automatic standby timers.

High Adaptability to Hardware Designs

□ **Platform:** Designed for Microchip's (formerly Atmel) AVR 8-bit microcontroller family, readily available version for the ATmega2560. The software may be ported to other microcontrollers by the developer.

□ **Display Support:** Supports 2 lines by 16 characters displays (HD44780 or compatible) or serial communication (using ANSI escape codes) at 57,600 baud.

□ **Flexibility:** Supports two different microcontroller pinouts for display and buttons. Buttons can be individual buttons or buttons that are interconnected via a resistor network.

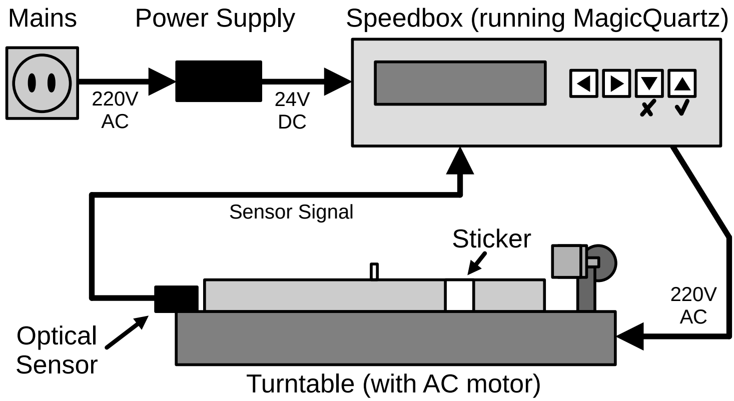
□ **Rapid Prototyping:** Support for widespread prototyping hardware, such as the *Arduino Mega 2560 Rev3*, including common peripherals (LCD keypad shields).

See part III on page 97 regarding hardware design considerations.

3. General Information

3.1. Typical Application

MagicQuartz is a microcontroller firmware that can be used for control and signal generation in a turntable speedbox. Such a speedbox could be powered with a low DC voltage from a fully isolated power supply. Usually, an inverter is integrated into the speedbox to generate the alternating current for the record player. The frequency of the AC is variable and is used to control the speed of the turntable. The speed can be measured, for example, via an optical sensor next to the record player's platter to which a paper sticker is attached:



Other configurations with other AC voltages, sensor types (e.g., magnetic) or construction forms (e.g., speed controllers built into turntables) are also conceivable (see part III on page 97). The instructions in this document assume that the MagicQuartz firmware has been installed on an external speedbox, similar to the one shown above.

3.2. Motor Compatibility

MagicQuartz is based on the principle that the AC frequency controls the speed of the turntable motor. Therefore, the approach only works with AC motors. However, as there are so many types, it cannot be guaranteed that a particular model will work with all MagicQuartz features. In general, however, the following applies:

- □ The "Live Quartz Mode" (see section 5.3 on page 40) with its small changes in AC frequency should work fine with all synchronous and asynchronous motors.
- □ If the motor relies on phase shift capacitors, greater changes in AC frequency should be avoided to prevent motor damage. It is then not recommended to use the speedbox function (see section 5.1.2 on page 35), as this changes the AC frequency in large steps. Note that it may be possible to modify and properly drive such a motor via a two-phase inverter configuration.
- □ Due to the slip inherent in asynchronous motors, the actual motor speed achieved may differ from the speed calculated by MagicQuartz, especially when using the speedbox or the transpose functions. However, it may

be possible to correct the speed in “Live Quartz Mode”. With such motors, voltage reduction usually does not work either, because the motor slows down with reduced voltage.

□ The hardware, in particular the inverter, must be laid out to meet to the motor’s power requirements.

3.3. About the Demo Version

A demo version of MagicQuartz is available for evaluation and hardware development purposes. It is fully functional, except for the following:

□ The standby timers `StandByOn` and `StandByOff` (see section 5.4.2.36 on page 64) cannot be disabled or set higher than 10 and 5 minutes, respectively. While this does not affect the playback of 45 and 78 RPM records, playback of 33 RPM records may be interrupted by the standby timer. Note that it is possible in any operational mode to prevent the software from entering standby without affecting playback by occasionally pressing a button on the speedbox.

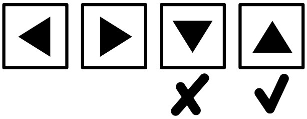
□ After some use of the software, MagicQuartz displays a nag screen and kindly asks you to register the software. The nag screen appears after the speedbox has been turned on and after waking up from standby.

These limitations can be overcome by registering the software. The registration process is described in section 5.4.2.10 on page 53.

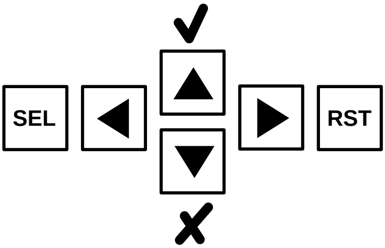
3.4. Keypad

3.4.1. Keypad Layouts

MagicQuartz is operated via a keypad with four or five buttons. The layout depends on the implementation of the hardware. The default layout is shown in the following figure on the left. If MagicQuartz is running on an Arduino or compatible development board, most likely an LCD keypad shield with the layout shown on the right is used. If the firmware is accessed via a serial connection, either the arrow keys or the number keys 1 to 4 (preferably the ones on the left side above the letter keys) on the computer keyboard can be used.



Default Keypad Layout



LCD Keypad Shield Layout

3.4.2. Keypad Functions

In general, the “Left” and “Right” buttons move the cursor on the screen. The “Up” and “Down” buttons increase and decrease the value under the cursor, respectively.

The “Up” and “Down” buttons also serve the following functions:

✓

□ **OK/Yes:** Accept what is shown on the screen.

□ **Enter:** Enter a function (such as the "Live Quartz Mode") or a menu entry (as in the "Configuration Menu").

□ **Clear Buffer:** In some functions, such as the "Live Quartz Mode", "Show Power Info", or "Show Sensor Info", pressing this button clears a buffer or temporary value.

✗

□ **Cancel/No:** Reject what is shown on the screen.

□ **Exit:** Exit a function (such as the "Live Quartz Mode") or a menu entry (as in the "Configuration Menu").

Please note that when using a serial connection, no graphical symbols can be shown. In some screens over a serial connection, the icons ✓ and ✗ are replaced with "`U`" and "`D`" to indicate using the "Up" and "Down" buttons, respectively. For more symbol mappings in serial mode, please refer to section 7.1 on page 79.

"Restart Inverter": By pressing the "Left" and "Right" buttons simultaneously, the "Restart Inverter" function (see section 5.4.2.2 on page 47) can be triggered. With an LCD keypad shield, the "Select" button has to be pressed. When a serial connection is used, the function can be triggered by entering "Shift+R" on the keyboard.

4. Getting Started

4.1. Flashing the Firmware

This section describes in detail how to perform the initial upload of the MagicQuartz firmware to an Arduino or compatible development board that employs the ATmega2560 microcontroller (please refer to section 14 on page 107 for supported hardware). The same procedure applies when performing future firmware updates. Note that any settings within MagicQuartz are kept when you follow these instructions.

The process described below is the same on all popular operating systems such as *Linux*, *Apple macOS* and *Microsoft Windows*. Although a *Linux* system is used in the following, the main differences when using *Windows* are highlighted.

4.1.1. Preparation

If you're starting with an assembled speedbox, disconnect the device from its power supply. Open the device and connect the development board to the computer via its USB interface, preferably using a galvanic USB isolator. Alternatively, you can remove the development board from the speedbox and then connect it to the computer via USB. If the speedbox doesn't integrate a development board, it may be possible to flash the firmware using an ISP programmer.

4.1.2. Software Installation

The firmware is distributed in the Intel HEX format. For flashing the firmware file, the open source software *avrdude* (see: `https://www.nongnu.org/avrdude/`) can be used. However, since using *avrdude* can be challenging, a good approach is to use the official *Arduino IDE* to obtain the command line parameters for *avrdude*.

Go to `https://www.arduino.cc/` and download and install the *Arduino IDE* for your operating system. The *Arduino IDE* already includes *avrdude*.

Note: When using *Windows* in combination with an Arduino-compatible development board, such as the "MEGA 2560 PRO", it may be necessary to install the "CH340" driver first. Please ask the seller or manufacturer of the development board for the driver.

4.1.3. Testing the Development Board

As a first step, it is necessary to configure the *Arduino IDE* for the correct development board. Start the *Arduino IDE* and select **Tools => Board: ... => Arduino Mega or Mega 2560**.

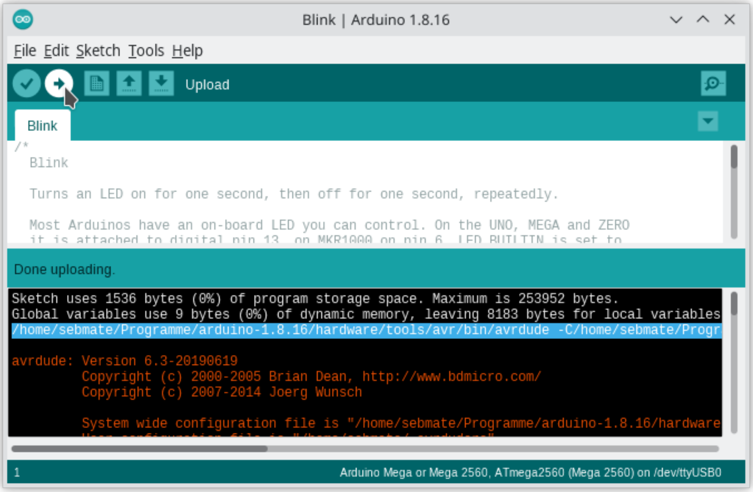
To check that everything works, you can flash a minimal program that is supplied with the *Arduino IDE*. Open the

“Blink” program via **File => Examples => 01.Basics => Blink**. Click on the upload button, which is the round button with the arrow pointing to the right (see next screenshot, under the mouse cursor). The *Arduino IDE* may ask you to select a serial port. Pick the one suggested and continue.

After the upload has succeeded, the LED on the development board should start blinking in one-second intervals.

4.1.4. Flashing the MagicQuartz Firmware

Go to the preferences of the *Arduino IDE*. Check the option “Show verbose output during: Upload”. Click on the “OK” button to go back to the main window. Upload the previous “Blink” program again. Notice the logging output in the lower section of the window:



```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
```

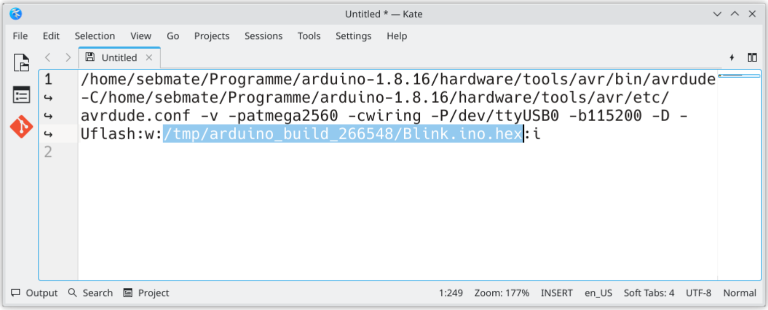
Done uploading.

```
Sketch uses 1536 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 8183 bytes for local variables
/home/sebmate/Programme/arduino-1.8.16/hardware/tools/avr/bin/avrdude -C/home/sebmate/Progr

avrdude: Version 6.3-20190619
         Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
         Copyright (c) 2007-2014 Joerg Wunsch

         System wide configuration file is "/home/sebmate/Programme/arduino-1.8.16/hardware
         User configuration file is "/home/sebmate/.avrduderc"
```

1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/ttyUSB0

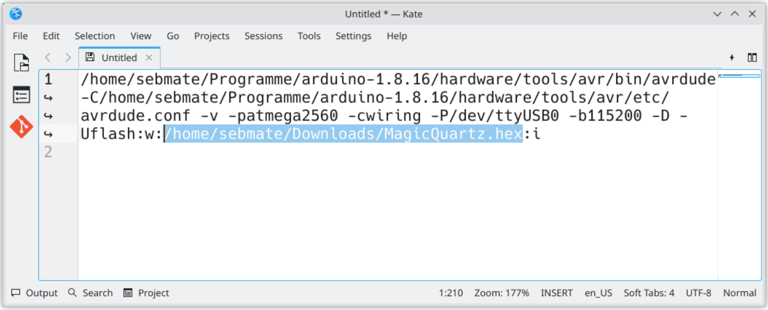
The third line (highlighted in blue in the screenshot above) is the command which was used to execute *avrdude*. Copy the full line into a text editor. Replace the file path pointing to the `Blink.ino.hex` file with the file path pointing to the downloaded `MagicQuartz.hex` file.

In my case, here's what the original command looks like:



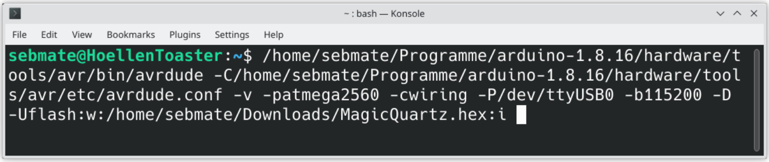
```
/home/sebmate/Programme/arduino-1.8.16/hardware/tools/avr/bin/avrdude
-C/home/sebmate/Programme/arduino-1.8.16/hardware/tools/avr/etc/
avrdude.conf -v -patmega2560 -cwiring -P/dev/ttyUSB0 -b115200 -D -
Uflash:w:/tmp/arduino_build_266548/Blink.ino.hex:i
```

And here is it with the path pointing to the `MagicQuartz.hex` file:



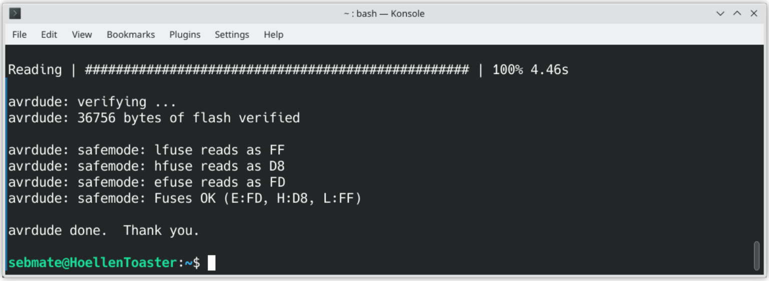
```
/home/sebmate/Programme/arduino-1.8.16/hardware/tools/avr/bin/avrdude
-C/home/sebmate/Programme/arduino-1.8.16/hardware/tools/avr/etc/
avrdude.conf -v -patmega2560 -cwiring -P/dev/ttyUSB0 -b115200 -D -
Uflash:w:/home/sebmate/Downloads/MagicQuartz.hex:i
```

Next, run the modified full command in a terminal to flash the MagicQuartz firmware to the development board (in *Windows*, use the *Command Prompt*, also known as **`cmd.exe`**):



```
sebmate@HoellenToaster:~$ /home/sebmate/Programme/arduino-1.8.16/hardware/tools/avr/bin/avrdude -C/home/sebmate/Programme/arduino-1.8.16/hardware/tools/avr/etc/avrdude.conf -v -patmega2560 -cwiring -P/dev/ttyUSB0 -b115200 -D -Uflash:w:/home/sebmate/Downloads/MagicQuartz.hex:i
```

The program's output should look similar to this:



```
Reading | ################################################## | 100% 4.46s

avrdude: verifying ...
avrdude: 36756 bytes of flash verified

avrdude: safemode: lfuse reads as FF
avrdude: safemode: hfuse reads as D8
avrdude: safemode: efuse reads as FD
avrdude: safemode: Fuses OK (E:FD, H:D8, L:FF)

avrdude done.  Thank you.

sebmate@HoellenToaster:~$
```

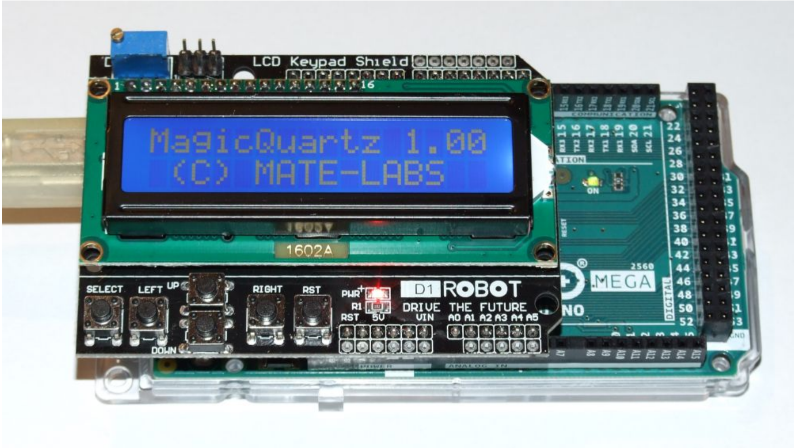
4.1.5. Verifying the Firmware Upload

4.1.5.1. Blinking LED

After the upload and whenever restarting the development board, its LED should blink very fast for one second. This indicates that the development board is now running the MagicQuartz firmware.

4.1.5.2. LCD Keypad Shield

If an LCD keypad shield is connected to the development board, you should see the firmware starting up, as in the following picture:



4.1.5.3. Serial Connection with GNU Screen (Linux/macOS)

If you're using a bare development board without an LCD keypad shield, you can interact with the firmware via the serial connection over USB using the *GNU Screen* program. To do this, run the following command in your terminal:

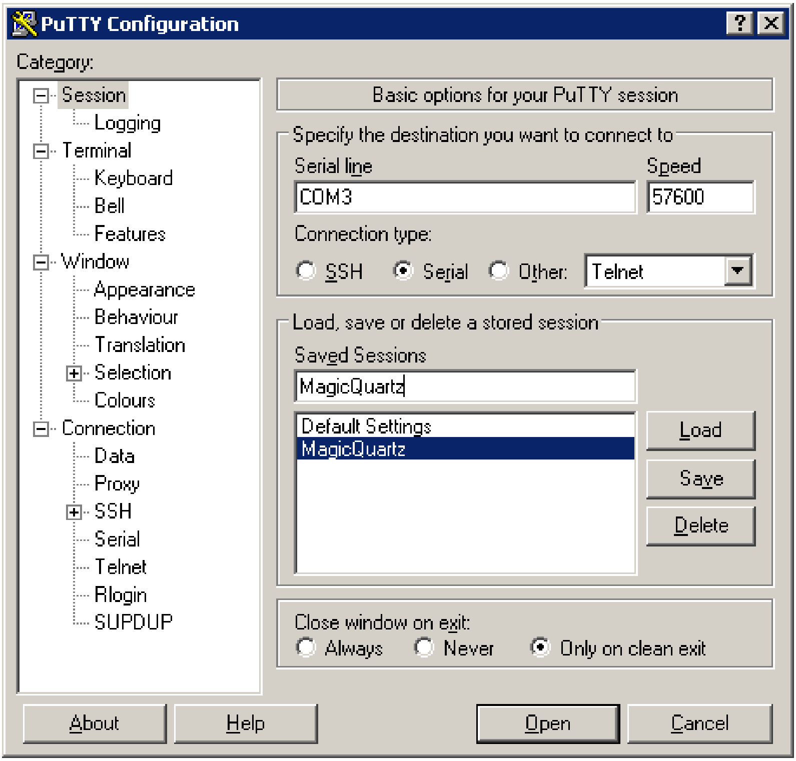
```
screen /dev/ttyUSB0 57600
```

You should then see MagicQuartz restarting (*GNU Screen* may automatically trigger a reset of the development board).

Note: If *GNU Screen* throws an error message, such as "Cannot exec '/dev/ttyUSB0': No such file or directory", you may have to use another TTY device. The easiest way to find out which one it is, is to review the setting in the Arduino IDE. It's shown under the menu **Tools => Board: ...**

4.1.5.4. Serial Connection with PuTTY (Windows)

In *Windows*, the program *PuTTY* (`https://www.putty.org/`) can be used to connect to the firmware. As shown in the following screenshot, enter the COM port, select the connection type “Serial”, and set the speed to 57600.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM3

Speed: 57600

Connection type: ○ SSH ● Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: MagicQuartz

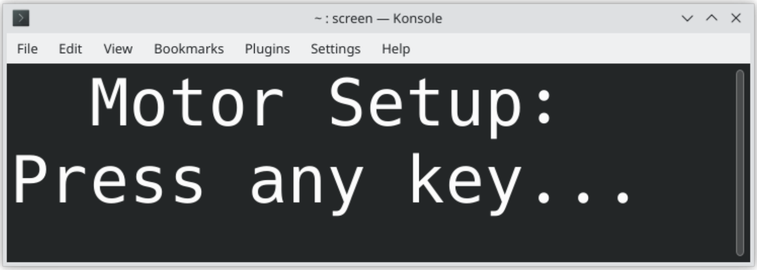
Default Settings
MagicQuartz

Close window on exit: ○ Always ○ Never ● Only on clean exit

It is also possible to save and load the configuration, as shown in the lower section of the screenshot. Please refer to the *PuTTY* documentation for details. Please be aware that the COM port’s number may change if multiple serial devices are connected to the computer.

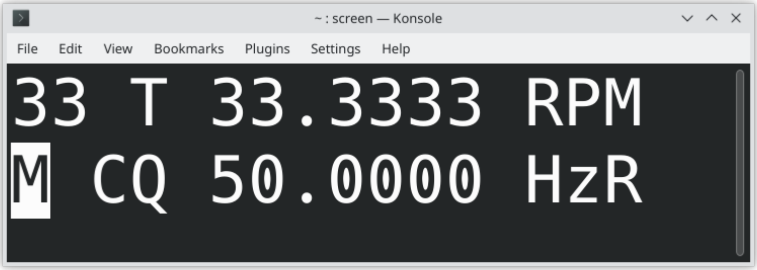
4.1.5.5. Using the Serial Connection

After startup, MagicQuartz will automatically ask you to setup the motor:



```
   Motor Setup:
Press any key...
```

Feel free to navigate yourself through the motor setup by using the arrow keys on your keyboard (for details on the motor setup, see section 4.2 on the facing page). After the motor setup, you will arrive at the main screen:



```
33 T 33.3333 RPM
M CQ 50.0000 HzR
```

To exit the *GNU Screen* program at any time, press “Ctrl+a”, followed by “k” and “y”. You may need to zoom out before you’re able to do this in case *GNU Screen* complains.

4.2. Motor Setup

Before the speedbox can be used with a turntable, it is necessary to properly configure the inverter for the connected record player. This ensures that the motor is driven with the correct voltage (e.g. 120 or 220 V). Although all parameters can also be determined and set manually, MagicQuartz features a setup wizard, called “Motor Setup”, which automates this process. Depending on whether the speedbox supports the “Advanced Power Management” (APM, see section 9 on page 91), the setup wizard also determines various other configuration parameters that are required for the current monitoring of the inverter.

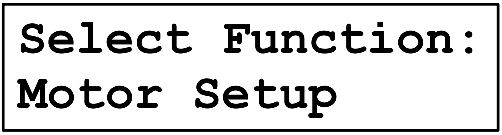
If the speedbox does not feature a built-in voltmeter, please use an energy cost meter or a true RMS multimeter to measure the generated voltage. In either case, observe the rules for safe handling of high voltages!

4.2.1. Starting the Motor Setup

The motor setup wizard is automatically started upon the first start of the firmware, after the flash memory has been formatted, or after the hardware has been upgraded with APM. In either case, MagicQuartz will display the following message:

```
 Motor Setup:
Press any key...
```

If you've previously completed or skipped this step, MagicQuartz will automatically go to the main screen. Don't worry: the motor setup can also be called from the configuration menu. Simply open the configuration menu, select the corresponding menu entry, and press the "OK" button:



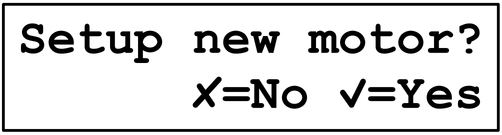
```
Select Function:
Motor Setup
```

For details on how to open and use the configuration menu, please refer to section 5.4 on page 46.

4.2.2. Motor Setup Walk-Through

MagicQuartz will now guide you through the motor setup and ask you to do different things, such as to set configuration parameters and to turn the motor on and off multiple times. The steps requested depend on whether the speedbox has APM available or not and whether MagicQuartz has been configured previously.

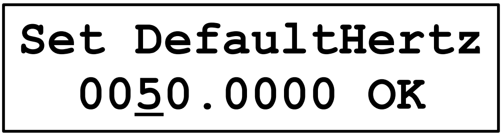
In case a motor has been set up previously, MagicQuartz may ask you if you aim to setup a new motor or not:



```
Setup new motor?
      ✗=No ✓=Yes
```

If you have connected a new or different turntable to the speedbox, answer this question with "Yes". Also answer it with "Yes", if you had problems with the motor setup previously. This resets the voltages to zero and thus prevents the risk of passing excessive voltage to the record player. If you, however, aim to refine the previous configuration using the same motor, you can select “No”, which will allow you to keep or modify the previous settings.

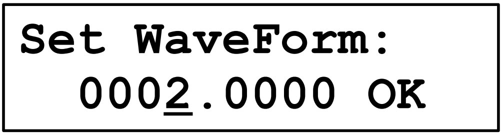
In the next step, MagicQuartz will ask you about the nominal frequency (in Hertz) of the turntable:



```
Set DefaultHertz
   0050.0000 OK
```

Set this to 50 Hz (as shown in the image above) or 60 Hz. This can be achieved by using the “Left” and “Right“ buttons to move the cursor below the digit to be changed and the “Up” and “Down” buttons to increase or decrease the digit. Move the cursor back to “OK” and press the “OK” button to accept the value. (For a general introduction on how to change parameters, see section 5.4.1 on page 46).

The next step is to set the desired wave form that MagicQuartz generates:



```
Set WaveForm:
   0002.0000 OK
```

Important: This value depends on the hardware of the speedbox. Please do not attempt to guess the value, instead, refer to the hardware documentation or section 8.3 on page 86.

If you’ve selected either wave form 1 or 2 in the previous step, MagicQuartz will also ask you about the phase shift:

```
Set PhaseShift:
  0000.0000 OK
```

Important: This changes the phase shift between channels 1 and 2. Note that the phase shift parameter may only be changed in dual-phase inverter configurations. For single-phase inverters, you must keep the value at 0. Setting other values in single-phase configurations may damage the inverter. For details please see section 8.3 on page 86.

The next step is to adjust the voltage with the load turned on (parameter `VoltageMtOn`, *Voltage Motor On*). For this, MagicQuartz asks you to turn the motor on and to adjust the voltage:

```
Turn motor on,
press OK button
```

```
Press OK button,
adjust voltage
```

The voltage is set on the following screen:

```
Set VoltageMtOn:
  0000.1000 OK
```

In contrast to the previous parameters, no whole number is to be entered here, but a floating point value between 0 and 1. The value affects the amplitude of the sine signal that is generated by MagicQuartz and consequently the voltage that is generated by the speedbox. 0 corresponds to no signal, 0.5 to half amplitude, and 1.0 to full amplitude. Since the value 1.0 produces the best signal (because the dynamic range is fully utilized), the following steps aim at getting as close as possible to this value.

Important: When increasing the value – as described below – it is recommended not to set it directly to 1.0, but to increase the decimal places step by step (e.g., 0.1, 0.2, 0.3, ...) to avoid unexpected large changes in the generated output voltage. Larger changes could lead to an over-voltage of the record player or to a shutdown of the inverter. If the inverter accidentally shuts down when you increase the voltage (in this case the voltage immediately drops to zero and the inverter may try but fail to turn back on), set the value back to 0.0 and slowly increase it again.

How to proceed depends on whether the inverter installed in the speedbox has a level control or not:

- □ If the inverter doesn't have a level control, slowly increase the parameter `VoltageMtOn` (via the decimal places, as described above) until the voltage generated by the inverter reaches the target voltage of the turntable (e.g. 220 V).

- □ If the inverter has a level control, turn the level control to zero. Set the `VoltageMtOn` value in MagicQuartz to 1.0. Because the level control is turned down, no voltage should be generated. Then very carefully turn up the level control on the inverter until the generated voltage reaches to the target voltage of the record player (e.g., 220 V). It may also be a good idea to set a slightly higher voltage at the inverter (e.g. 225

V, while `VoltageMtOn` = 1) and then to reduce the `VoltageMtOn` value again (e.g., to 0.97) so that the desired voltage (220 V) is accurately obtained. This leaves some room for increasing the voltage later in software.

If APM is not available, the process is now finished and “Done!” is shown. If APM is available, the motor setup continues with requesting setting `VoltageMtOff` (*Voltage Motor Off*) and automatically determining the parameters `MotorRunThrs`, `MotorSwThrs`, `OvlProtThrs` and `InvIdleCurr`, as shown in the following:

```
Turn motor off,
press OK button
```

```
Stop platter,
press OK button
```

```
Press OK button,
adjust voltage
```

On the next screen, adjust the voltage (for the motor being switched off) by slowly increasing the `VoltageMtOff` value:

```
Set VoltageMtOff
  0000.1000 OK
```

MagicQuartz then automatically determines some parameters:

```
MotorRunThrs=254
MotorSwThrs=203
```

```
InvIdleCurr=45
```

Finally, MagicQuartz asks you to turn the motor on once more:

```
Turn motor on,
press OK button
```

```
OvlProtThrs=540
```

And the process is finished:

```
Done!
```

4.3. Testing the Motor Setup

If the motor setup was successful, the speedbox should now work properly. Nonetheless, it is a good idea to double-check this. This section briefly describes how to do this. In case problems occur, please refer to section 6 on page 67. Note that it may also help to execute the “Motor Setup” function again, as described in the section above.

- □ **General Motor Status Detection:** On the main screen, switch the turntable motor on and off. The motor status indicator in the lower right corner should show a flash symbol (or an an “R” when using a serial connection) whenever the motor is switched on and show nothing when the motor is switched off. If this is not working, please refer to section 6.1.1 on page 67.

- □ **Motor Status Detection upon Startup:** It is possible that the motor is already switched on when the speedbox is starting up. Verify that MagicQuartz is able to correctly detect this. If this is not the case, please refer to section 6.1.2 on page 68.

- □ **Generated Voltages:** Check that the generated supply voltage stays close to the rated voltage (e.g., 220 V) when the motor is switched on and off. Small voltage fluctuations (about 5 V, depending on the inverter hardware) are normal because MagicQuartz needs a very short time to correct the voltage (about 200 ms). If the voltages are off, please refer to section 6.1.3 on page 69.

- □ **Overload Detection:** Connect a load that slightly exceeds the rating of the record player and turn it on. MagicQuartz should then display an overload warning, as described in section 6.2.2 on page 72. If this is not working, please refer to section 6.1.4 on page 69.

- □ **Inverter Shutdown Detection:** MagicQuartz may be able to detect if the inverter has shut down due to a fault, but this depends on the hardware implementation. In particular, if the inverter's idle current draw (no AC is being generated) is very close to the current draw when it is generating an AC with no motor load, detection may not work. The shutdown detection can possibly be triggered by manually switching off the inverter (if the hardware implementation allows this) or by setting a too low `RampingSpeed` value which causes the inverter to switch off. If the detection is working, MagicQuartz should display a message warning about the inverter having shut down, as described in section 6.2.3 on page 73.

4.4. Next Steps

The setup is now complete and the speedbox may now be used. Nevertheless, here are a few more tips on what you could do next:

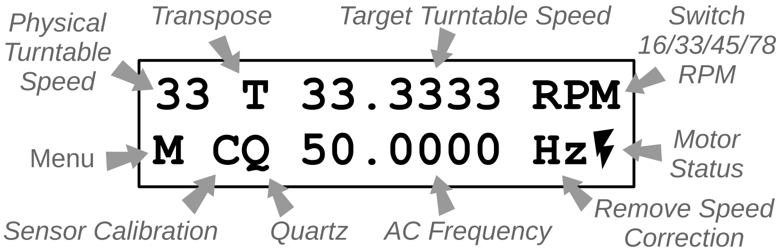
- □ Use the "Sensor Calibration Mode" (see section 5.2 on page 37) to properly setup the motion sensor. After this, try "Live Quartz Mode", as described in section 5.3 on page 40.
- □ If the sensor works fine, you may set `QuickCalib` to 1 to speed up entering "Live Quartz Mode" next time (see section 5.4.2.25 on page 60).
- □ If the inverter has warmed up in the meantime, the voltages may have drifted slightly. You can correct the voltages as described in section 5.4.2.14 on page 55.
- □ Depending on the motor, reducing the operating voltage may significantly reduce motor noise, which is pos-

sible via the **`VoltageMtRed`** and **`ReducVoltage`** (sections 5.4.2.15 and 5.4.2.16 on page 56) parameters.

5. Menus and Functions

5.1. Main Screen

The main screen allows changing the core speed settings for the connected turntable. It also acts as the entry point to all other functions (e.g. the configuration menu and the "Live Quartz Mode"). It is cursor-controlled: use the "Left" and "Right" buttons to move the cursor below the digits representing values (e.g. `33.3333` or `50.0000`) or characters representing menus or functions (e.g. "`M`", which enters the configuration menu). Then use the "Up" and "Down" buttons to change the values or to enter the menu or function.



Physical Turntable Speed: Set this to match the physical speed setting of the record player. "Physical" means the belt or idler wheel setting on the turntable. For example, if the record player is set to play at 45 RPM, also set the physical platter speed in MagicQuartz to 45. Use the “Up” and “Down” buttons to switch between the standard record speeds (16, 33, 45, and 78 RPM). Note that this setting also changes the “Target Turntable Speed” to the respective speed, which in general does not yet change the AC frequency. However, if the speedbox was previously generating a non-nominal motor AC frequency, it is ramped back to the nominal frequency.

Transpose: Use the “Up” and “Down” buttons to transpose the playback in semitones.

Target Turntable Speed: Move the cursor below the desired digit, then use the “Up” and “Down” buttons to increase or decrease the RPM. The unit automatically calculates the corresponding AC frequency.

Switch 16/45/45/78 RPM: This allows direct switching between the standard record speeds. As such, it provides access to the speedbox function as described in section 5.1.2 on the facing page. Move the cursor below “`RPM`” and use the “Up” and “Down” buttons to switch between the speeds. It is highly recommend to read section 5.1.2 in advance.

Motor Status: This shows if the turntable’s motor is running, or not. A flash symbol (or an “`R`”) is displayed whenever it is running, otherwise nothing is displayed.

Remove Speed Correction: Move the cursor below “`Hz`” and press the “OK” button to switch back to the default AC frequency (50 or 60 Hz). This deletes the current speed correction factor, if there is any. The speed correction factor is automatically determined in “Live Quartz Mode” (see section 5.3 on page 40) and kept in the other modes, such as the main screen.

AC Frequency: Move the cursor below the desired digit. Then use the "Up" and "Down" buttons to increase or decrease the AC frequency. The software automatically calculates the corresponding target turntable speed.

Quartz: Enters the "Live Quartz Mode", as described in section 5.3 on page 40.

Sensor Calibration: Enters the "Sensor Calibration Mode", as described in section 5.2 on page 37.

Menu: Enters the "Configuration Menu", as described in section 5.4 on page 46.

5.1.1. Changing the RPM or Frequency

Move the cursor with the "Left" and "Right" buttons below one of the digits (those of `33.3333` or `50.0000`) and use the "Up" and "Down" buttons to increase or decrease the value. If you change the RPM, MagicQuartz will automatically calculate the corresponding AC frequency and vice versa. The playback speed of the record player changes accordingly.

5.1.2. Speedbox Function

The speedbox function allows changing the playback speed of the turntable without changing the physical speed setting on it. This may be useful if the record player does not provide an easy way to change the playback speed. For example, some modern turntables require the user to move the belt to a different pulley below the platter in order to switch between 33 and 45 RPM. With the speedbox function, this can be done electronically. It may also enable playing records with speeds that a record player normally does not support.

The general approach is:

1. Change the physical platter speed in the upper left corner to match the current setting of the turntable.
2. Move the cursor below “`RPM`” in the upper right corner and use the “Up” and “Down” buttons to directly switch between the standard record speeds.

Important notes:

□ Using the speedbox function changes the generated AC frequency in large steps. For example, switching to 78 RPM with a physical setting of 33 RPM raises the AC frequency to 117 Hz (for a 50 Hz unit). This could damage the record player’s motor – in particular, if it relies on phase shift capacitors.

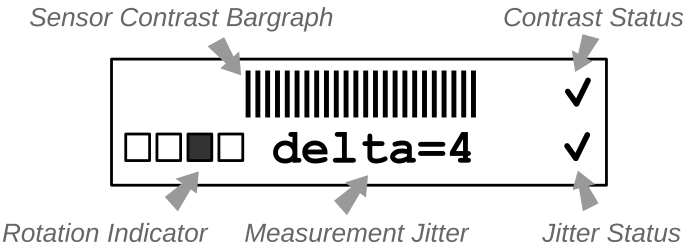
□ It is recommended that the turntable is set to a speed that is close to the desired speed. For example, if you want to play back a 78 RPM record, don’t use 33 RPM as the physical platter speed, instead, use 45 RPM if the record player supports it. This reduces the frequency difference between the nominal frequency of the motor (typically 50 or 60 Hz) and the frequency required to obtain the target speed.

□ The AC frequency is a calculated value and may not be optimal, depending on the type of the turntable motor. This may be especially the case if the record player is equipped with an asynchronous motor. After electronically switching to a different speed, use the “Live Quartz Mode” (see section 5.3 on page 40) to precisely determine the optimal AC frequency.

5.2. Sensor Calibration Mode

5.2.1. Overview

In “Sensor Calibration Mode”, MagicQuartz automatically adjusts itself to the dynamic range of the motion sensor, which is used to determine the speed of the platter. In most cases this works without problems. However, a properly positioned motion sensor can improve the performance of MagicQuartz. For this purpose – and in the unlikely event that problems occur – the sensor calibration mode can help optimizing the position of the motion sensor.



The **Rotation Indicator** in the lower left corner is used to visualize whether MagicQuartz is able to correctly detect the rotation of the turntable’s platter, or not. At each rotation, the indicator should progress the next rectangle. If the detection is erroneous, the rotation indicator jumps or skips. As a result, it also displays erratic, sometimes high delta values (see below).

When using an optical motion sensor, highly reflective platters can cause such problems. In combination with an unevenly machined platter rim, an oscillating reflection can trigger false edges in the measurement signal. This can be circumvented by slightly tilting the sensor to prevent it from picking up this reflection.

Sensor Contrast Bargraph and **Contrast Status:** The bargraph on top visualizes the measured sensor image contrast. Try to increase the sensor contrast as much as possible (the bargraph on top then becomes longer). A higher contrast allows MagicQuartz to better detect the signal edge of the bypassing paper sticker on the record player's platter. You can influence the contrast by moving the sensor closer or further off to the spinning platter. It takes some time until the bargraph changes, depending on the current RPM setting. ***Note:*** It is normal that the bar's ends sometimes can move by three vertical lines on one or both sides.

If the contrast is good enough, a check mark (✓) is displayed on the right side next to the bar, as shown in the image above. If otherwise a problem is detected, MagicQuartz automatically displays the following recommendations:

↧ **Increase Distance:** The sensor is too close to the platter of the turntable. Increase the distance between the sensor and the platter to improve the contrast.

⤒ **Reduce Distance:** The sensor is either not connected or too far away from the platter of the record player. Connect the sensor or reduce the distance between the sensor and the platter of the turntable to improve the contrast.

Low Contrast: Check if you have attached a paper sticker to the record player's platter and if the platter rotates. If everything is set up correctly but still does not work, it may help to use a sticker with a different color (e.g. black instead of white). In general, the sticker should be of a different shade in comparison to the turntable's platter.

5.2.2. Understanding the Measurement Jitter

The **Measurement Jitter**, as indicated by the delta value in the second line of the display, is the deviation in centimilliseconds between two measurements. A centimillisecond is 100 µs, which is the internal timing resolution of MagicQuartz. If the platter rotates at exactly 33.3333 RPM, a full platter revolution takes exactly 1.8 seconds, 1,800 milliseconds, or 18,000 centimilliseconds. However, due to measurement errors (signal noise, ambient light flicker, etc.), MagicQuartz typically measures some variations, such as 18,002, 17,998, 18,000, 18,001, 17,997, and so on. For the first two measurements, the delta value would be 4 (the difference between 18,002 and 17,998).

To improve the overall performance of MagicQuartz, it is desirable to keep this jitter as low as possible. The ideal value would be zero, which is usually never reached in the long run. However, lower values (below 20) are acceptable because MagicQuartz' speed regulation algorithm is able to deal efficiently with this type of measurement error.

When an acceptable jitter value is measured over a certain time, MagicQuartz displays a check mark (✓) right to the delta value. If the jitter is too high (e.g. > 20), a cross (✗) is displayed instead.

You can try to reduce the measurement jitter by changing the position of the optical motion sensor. Unfortunately, higher contrast does not necessarily mean lower jitter. Moving the sensor closer to the platter (and thus reducing contrast in some cases) may also help. This is where some experimentation is needed. Avoiding fluorescent or changing ambient light is also a good idea.

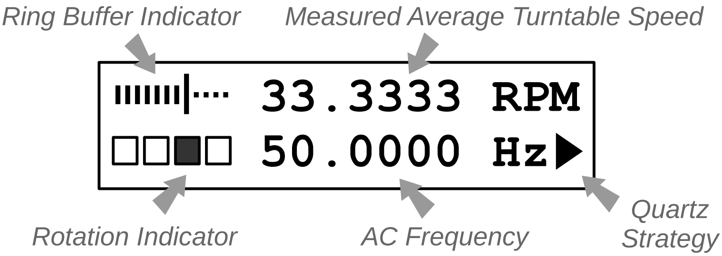
MagicQuartz also features a software low-pass filter for the optical motion sensor that can be set via the `SensLowPass` parameter (see section 5.4.2.24 on page 60). You can also change the strength of the low-pass filter directly in sensor calibration mode using the "Left" and "Right" buttons. A higher value means stronger filtering. This reduces the signal noise, but also reduces the edge quality of the bypassing sticker. You can experiment with this parameter to further improve the jitter.

5.3. Live Quartz Mode

5.3.1. Overview

MagicQuartz features the so called "Live Quartz Mode", in which it constantly monitors and corrects the record player's speed. Based on the current speed, the optimal AC frequency is calculated and generated to correct possible speed deviations.

To enter this mode, select the “`Q`” symbol in the bottom left corner of the main screen and press the “OK” button. MagicQuartz briefly enters the “Sensor Calibration Mode” (see section 5.2 on page 37) before it continues into “Live Quartz Mode”. Press the “Cancel” button to exit at any time. The “Live Quartz Mode” screen is depicted in the following figure:



The **Rotation Indicator** has the same functionality as in “Quartz Calibration Mode” (see section 5.2 on page 37 for details). It should progress to the next rectangle at each rotation and never skip.

MagicQuartz displays the **Measured Average Turntable Speed** of the turntable’s platter on top and the computed and generated **AC Frequency** on the bottom of the screen.

Pressing the “OK” button clears the measured average record player speed. Note that this does not affect the speed regulation. Keeping the “OK” button pressed enters a temporary “fast” mode with `QrtzLatency` = 1 (see section 5.3.4 on page 45) and an “`F`” is shown in the lower right corner. This enables MagicQuartz to quickly adopt to speed changes.

The generated AC frequency has to be higher at the beginning of a record than at the end of a record. In order to quickly arrive at the ideal frequency when a new record is put on, MagicQuartz is capable of automatically storing and restoring this frequency. Whenever the frequency is saved at the beginning of a record, "`-SAVED-`" is displayed. When "Live Quartz Mode" is entered, or when the motor has been switched off and on again in "Live Quartz Mode", "`(Speed restored)`" is displayed, and the initial AC frequency is restored. The time at which the frequency is saved can be configured via the parameter `SaveHzAtSec`, as described in section 5.4.2.35 on page 64.

5.3.2. Understanding the Speed and Frequency Values

The measured average turntable speed is an unweighted moving average of the previous speed measurements (one per rotation). MagicQuartz calculates this speed by storing speed measurements in a ring buffer. When this buffer is full, the oldest values are overwritten again. The fill level and current writing position are displayed with the **Ring Buffer Indicator** in the upper left corner of the screen. The buffer size is determined by the `BufferSize` parameter, which can be changed in the configuration menu. Depending on how the parameter is set, the measured average speed display reacts to speed changes more promptly (smaller values for `BufferSize`) or less promptly (larger values for `BufferSize`). When MagicQuartz decides it is necessary, the buffer is automatically cleared (see parameter `BufferClear`). At any time in "Live Quartz Mode", press the "OK" button to clear the ring buffer.

If MagicQuartz is not able to keep up with the speed regulation (see section 5.3.3 on the next page), or whenever it does not make sense to average the speed measurements, “`LIVE`” is displayed instead of the graphical indicator and the currently measured speed is shown.

MagicQuartz computes the AC frequency using a weighted moving average algorithm (the weight is the parameter **`QrtzLatency`**, see section 5.3.4 on page 45), which takes into account the quality of each measurement. As a result, the speed regulation algorithm can deal with measurement errors efficiently. It is completely independent from the previously described “simple” average speed computation. As a result, the measured average record player speed cannot be used to evaluate the performance of the speed regulation algorithm. This can be done with the speed regulation symbols described in the next section.

5.3.3. Speed Regulation Symbols

MagicQuartz displays icons in the lower right corner which can be used judge how well the speed regulation works.

▶ **Perfect Speed:** MagicQuartz considers the current speed of the turntable as equivalent to the target speed of the record player.

The following symbols may keep appearing during normal operation. This is perfectly fine as long as the “Perfect Speed” symbol from above keeps appearing regularly:

▲ **A Bit Too Slow:** MagicQuartz is considering the current platter speed a bit too low and has been up-regulating the AC frequency.

▼ **A Bit Too Fast:** MagicQuartz is considering the current platter speed a bit too high and has been down-regulating the AC frequency.

⏫ **Too Slow:** MagicQuartz is considering the current platter speed as too low and is constantly up-regulating the AC frequency. If this symbol does not disappear, consider reducing the `QrtzLatency` parameter (see section 5.3.4 on the next page).

⏬ **Too Fast:** MagicQuartz is considering the current platter speed as too high and is constantly down-regulating the AC frequency. If this symbol does not disappear, consider reducing the `QrtzLatency` parameter (see section 5.3.4 on the facing page).

⏸ **Paused:** MagicQuartz has paused the speed regulation. This for example happens after MagicQuartz has rejected bad measurements (see next) or waits for an accelerating turntable platter. For details, see parameter `WaitOnBad` and `WaitOnStart` (see sections 5.4.2.32 and 5.4.2.33 on page 63).

✖ **Dropped:** MagicQuartz has detected and dropped a bad measurement. This usually happens on automatic record players when the automatic stop function kicks in at the end of a record. For details, see parameter `BadDelta` (see section 5.4.2.34 on page 63).

In addition, the following symbols inform about special conditions:

■ **Stopped:** The speed regulation is turned off (`QrtzLatency` = 0, see section 5.3.4 on the next page).

LIM **Limit Reached:** The limit of the speed regulation range has been reached and MagicQuartz has stopped regulating the AC frequency for safety reasons. Please refer to the description of the parameter `QuartzRange` (section 5.4.2.31 on page 62) for more information.

5.3.4. Tuning the Speed Regulation

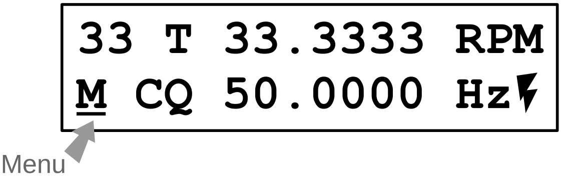
The speed regulation can be tuned via the **`QrtzLatency`** parameter (see section 5.4.2.30 on page 62). This can either be set in the configuration menu or by pressing the “Left” and “Right” buttons in “Live Quartz Mode”.

Setting **`QrtzLatency`** to 0 stops the speed regulation. A value of 1 sets the speed regulation to full strength, which allows MagicQuartz to quickly respond to speed changes. The drawback is that this also causes larger AC frequency changes. In contrast, setting **`QrtzLatency`** to higher values reduces the strength of the speed regulation.

As explained in section 5.3.3 on page 43 for the double-arrow symbols, the speed regulation may not be able to keep up with the speed drift if **`QrtzLatency`** is set too high. MagicQuartz then constantly shows double up or down arrows. If this happens, simply reduce **`QrtzLatency`**. It is recommended to set **`QrtzLatency`** as high as possible to avoid larger AC frequency changes, but low enough to let the speed regulation algorithm properly keep up with speed changes. The default value is 3. Values higher than 6 are typically not recommended.

5.4. Configuration Menu

The configuration menu allows you to set up MagicQuartz. It allows you to execute different functions and to set parameter values. The configuration menu can be accessed from the main screen by selecting the “`M`” entry in the lower left corner:

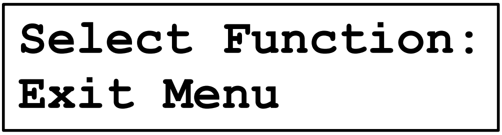


```
33 T 33.3333 RPM
M CQ 50.0000 Hz⚡
```

Use the “Left” and “Right” buttons to move the cursor below the “`M`” and press the “OK” button.

5.4.1. General Use

The configuration menu starts with the “Exit Menu” entry:

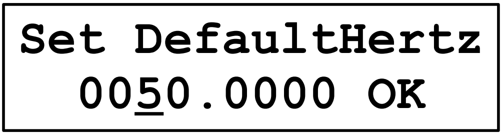


```
Select Function:
Exit Menu
```

Use the “Left” and “Right” buttons to switch between the menu entries. Press the “OK” button to enter a function. To exit the menu, either select the “Exit Menu” entry (as shown above), or press the “Cancel” button in any menu position.

All entries starting with “`Set ...`”, as shown in the screen below, change numeric configuration parameters. Within these, use the “Left” and “Right” buttons to select a digit.

You can then increase or decrease the value by pressing the “Up” and “Down” buttons. To exit the screen, move the cursor below “OK” and press the “OK” button.



```
Set DefaultHertz
 0050.0000 OK
```

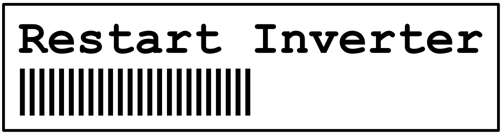
5.4.2. Configuration Menu Entries

5.4.2.1. Exit Menu

Exits the configuration menu and returns to the main screen. Note that the menu can also be exited from any other position by pressing the “Cancel” button.

5.4.2.2. Restart Inverter

As the name suggests, this function restarts the inverter. This is done by first reducing the voltage to zero and then increasing it again. Normally, and if APM is available, MagicQuartz will detect by itself when the inverter has shut down and inform the user. In other cases, the inverter can be restarted manually using the “Restart Inverter” function.



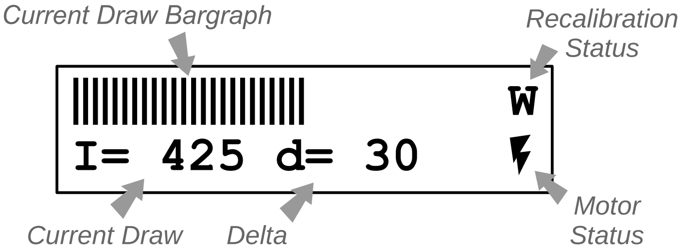
```
Restart Inverter
```

The function can also be triggered in most of the screens by pressing the “Left” and “Right” buttons simultaneously. When using an LCD keypad shield, the same can be attained by pressing the “Select” button. When using a serial connection, the function can be triggered by entering “Shift+R” on the computer keyboard.

Note: If you use the function in “Live Quartz Mode”, the `QrtzLatency` value may change unintentionally, if both buttons are not pressed at exactly the same time.

5.4.2.3. Show Power Info

This function displays information about the inverter’s current draw, which can be used to calculate various other configuration parameters. It displays the internal measurement of the current draw, which is displayed in the lower left corner. This function requires APM.



The **Current Draw** by the inverter is shown in the lower left corner. ***Note:*** The range is from 0 (no current is drawn) to 1023, which is the theoretical upper value that can be measured by the microcontroller. The true upper value may be, depending on the hardware implementation, lower.

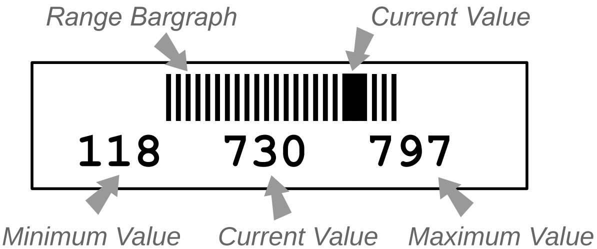
Switching the motor on and off should display different current values, and the **Motor Status** indicator should change accordingly (if not, you may need to calibrate the motor or change MotorSwThrs). As in the main menu, a flash symbol is shown in the lower right corner when MagicQuartz believes that the motor is switched on.

The “Show Power Info” function also displays a temporary **Delta** value, which can be zeroed by pressing the “OK” button.

The inrush current compensation behavior (see section 9 on page 91) is shown with the **Recalibration Status**, which displays a “W” (“wait”) while waiting MotorSwTime, and a “C” (“calibrate”) when measuring the stable current again. At nominal motor frequency, an “S” (“simple”) is shown instead to indicate that the MotorRunThrs value is used to determine the motor running status.

5.4.2.4. Show Sensor Info

This function displays sensor information about the motion sensor. This function displays the measurement value of the optical sensor. The range is from 0 to 1023.

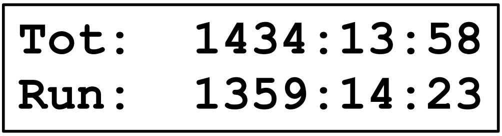


The three numbers on the bottom display the minimum, current and maximum values. The current value is updated at a lower refresh rate (to be readable), so quick changes may not be shown. It might be good idea to slowly turn the platter by hand. All values can be reset by pressing the “OK” button.

5.4.2.5. Show Statistics

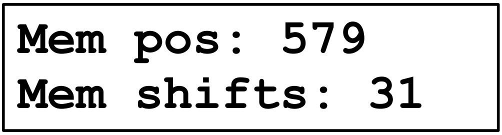
This function displays various usage statistics, as explained in the following.

On the first screen, the first line displays the total time the unit has been switched on. The second line displays the “running” time, which is the time the turntable motor was switched on. The purpose of this is to keep track the wear of the record player stylus.



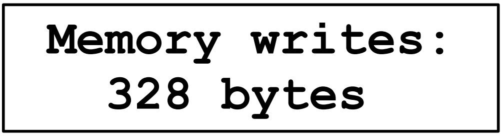
```
Tot:  1434:13:58
Run:  1359:14:23
```

Pressing the “OK” button displays additional information on the flash wear leveling. This includes the currently used memory position and the count of memory shifts:



```
Mem pos: 579
Mem shifts: 31
```

Pressing the “OK” button again displays the number of bytes that had been written to the flash memory since the unit has been started up:



```
 Memory writes:
   328 bytes
```

Pressing the "OK" button once more displays the time when the next memory shift has to take place:

```
Next mem shift
  due in 18h
```

MagicQuartz automatically shifts its flash memory every 24 hours of operation to spread memory writes across the whole memory area. For details on the flash memory handling, please see section 11 on page 95.

5.4.2.6. Motor Setup

This function, which is described in detail in section 4.2 on page 23, is used to setup the inverter and the turntable motor by requesting the user to set `VoltageMtOff`, `VoltageMtOn`, and automatically determining the parameters `MotorRunThrs`, `MotorSwThrs`, `OvlProtThrs`, and `InvIdleCurr`. It also resets `VoltageMtRed` and `ReducVoltage`.

5.4.2.7. Calib OvlProtThrs

This function determines the parameter `OvlProtThrs` (*Overload Protection Threshold*). While this is also done in the "Motor Setup" function (see above), the "Calib OvlProtThrs" function follows a different approach. It aims to determine the maximum current that can be measured, whereas "Motor Setup" simply picks a value that is slightly above the turntable motor's maximum startup current.

To use this function, simply connect a load that exceeds the speedbox' rating, such as a 60W light bulb, to the speedbox. ***Note:*** This may blow the fuse of the speedbox.

```
Turn on load...
OvlProtThrs=520
```

Press the "OK" button to exit this function again.

5.4.2.8. Format Memory

This function formats the microcontroller's internal flash memory and effectively resets the firmware to "factory" default. After the setup, the "Motor Setup" has to be executed again. Before executing this function, please make sure you have your registration code to hand; if not, please recover it from the "Register" function (see section 5.4.2.10 on the facing page). It is also a good idea to take note of the two clock timers `TotalHours` and `PlayHours` (see section 5.4.2.37 on page 65) so that these can also be restored after the reset.

5.4.2.9. Shift Memory

MagicQuartz automatically shifts the whole flash memory by one byte every 24 hours to distribute the write operations across the whole memory area. This function triggers the memory shift manually and resets the timer back to 24 hours. More information about the flash wear leveling can be found in section 11 on page 95.

5.4.2.10. Register

This function allows you to register your MagicQuartz license to unlock all software features. The “Register” function will display two hardware IDs, which are unique to your ATmega2560 microcontroller and look similar to this:

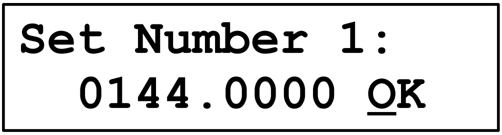
```
Hardware ID 1:
 1F45-EB43-E451
```

```
Hardware ID 2:
   4711-FE8C
```

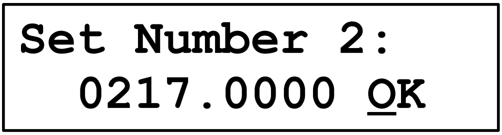
When registering via email, please send these two hardware IDs to Sebastian Mate (MATE-LABS). You will receive a registration certificate that contains an activation code comprising ten numbers, similar to this one:

```
Hardware ID: 1F45-EB43-E451 4711-FE8C

Activation Code: 144 217 163 171 10 77 199
152 0 103

Number 1: 144
Number 2: 217
Number 3: 163
Number 4: 171
Number 5: 10
...
Number 9: 0
Number 10: 103
```

In the subsequent screens, enter the numbers 1 to 10 as illustrated in the following:

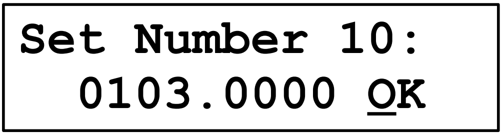


```
Set Number 1:
  0144.0000 OK
```



```
Set Number 2:
  0217.0000 OK
```

. . .



```
Set Number 10:
  0103.0000 OK
```

MagicQuartz will then attempt to validate the activation code. If this process was successful, the license is registered afterwards. ***Thank you for registering!***

5.4.2.11. Set WaveForm

This function is used to set the wave form that is generated by the inverter. Please refer to section 8.3 on page 86 for details.

Default: *0*

Range: *0–3*

5.4.2.12. Set PhaseShift

This function is used to set the optional phase shift. The setting is only applicable when `WaveForm` is 1 or 2. Please refer to section 8.3 on page 86 for details.

Default: *0*

Range: *0–255*

5.4.2.13. Set DefaultHertz

Sets the default (nominal) AC frequency. It should match the mains frequency the turntable was built for (50 or 60 Hz).

Default: *50*

Range: *40–70*

5.4.2.14. Set VoltageMtOff / VoltageMtOn

The parameters `VoltageMtOff` (*Voltage Motor Off*) and `VoltageMtOn` (*Voltage Motor On*) affect the generated output voltages when the motor is switched on or off, respectively. When changing one of these voltages, make sure the motor is in the appropriate switched state (i.e., switched on if you change `VoltageMtOn`). To avoid unexpected large changes in the output voltage, it is advised to change the lower digits first.

The voltage rises as the value rises. Setting the value to 1 results in the highest possible output voltage while preserving a perfect sine wave. When the value is greater than one, the generated sine wave is chopped off at its peaks, and humming from the inverter may be audible. The generated wave form changes to a rectangular wave at higher settings. As a result, it is advised to keep the values below or equal to 1.

If you can't get a sufficient output voltage even with the values set to 1, the speedbox may not be suitable for driving the record player model.

Range: *0–9999.9990 (it is advised not to go above 1)*

5.4.2.15. Set VoltageMtRed

The parameters `VoltageMtRed` (*Voltage Motor Reduced*) affects the voltage that is is generated in the reduced voltage mode in "Live Quartz Mode".

Reducing the voltage can lead to a significant reduction of motor noise (i.e., hum caused by vibration) in some turntable motors, but it does not work with all motors. If the record player slows down when the voltage is reduced, this feature is not suitable for the motor.

For the voltage to be reduced in "Live Quartz Mode", the parameter `ReducVoltage` (see below) must be set to 1.

Range: *0–9999.9990 (it is advised not to go above 1)*

5.4.2.16. Set ReducVoltage

Setting this parameter to 1 enables the automatic voltage reduction in "Live Quartz Mode".

Default: *0*

Range: *0–1*

5.4.2.17. Set RampingTime

The `RampingTime` parameter affects how fast the voltage is decreased and increased when the inverter is shutting down, starting up, or restarting. The value roughly corresponds to the up-ramping time in milliseconds and the down-ramping time in 0.5 milliseconds.

The value should be set as low as possible for performance reasons, but high enough to prevent the inverter from unintentionally entering a protection state during start-up. The default value was chosen quite conservatively (i.e., high) to improve out-of-the-box hardware compatibility.

Default:	*250*
Range:	*0–9999*

5.4.2.18. Set MotorRunThrs

The `MotorRunThrs` (*Motor Running Threshold*) parameter sets the motor's current draw above which the motor is considered to be switched on upon startup. The value is also used to detect the motor switching status at nominal AC frequency +/– 2 Hz. With a value of 0 MagicQuartz assumes that the motor is always switched on. Please refer to the section 9 on page 91 for details.

Default:	*310*
Range:	*0–1023*

5.4.2.19. Set OvlProtThrs

The `OvlProtThrs` (*Overload Protection Threshold*) parameter sets the motor's current draw at which the amplifier is supposed to be overloaded during normal operation. Please refer to the section 9 on page 91 for details.

Default: *600*

Range: *0–1023*

5.4.2.20. Set InvIdleCurr

The `InvIdleCurr` (*Inverter Idle Current*) parameter specifies the expected quiescent current of the inverter. It is used to detect possible inverter shutdowns (which may happen due to the inverter overheating and shutting down). It is also used upon boot to detect a possible internal short-circuit of the inverter, if the current measured is greater than twice `InvIdleCurr`. Please refer to section 9 on page 91 for details.

Default: *0*

Range: *0–1023*

5.4.2.21. Set ProtectDelay

The `ProtectDelay` (*Protection Delay*) parameter sets the time delay after which an inverter shutdown (inverter current less than `InvIdleCurr`) or overload (inverter current higher than `OvlProtThrs`) is handled by the software. The delay is calculated from the set value multiplied by 200 ms, i.e. 3 corresponds to 0.6 seconds.

It is recommended to set a value to the lowest possible value that does not yet produce false error messages. At the same time, the value should be low enough to prevent a premature automatic restart of the inverter (which often fails depending on the hardware design).

Default: *3*

Range: *0–9999*

5.4.2.22. Set MotorSwThrs

The `MotorSwThrs` (*Motor Switching Threshold*) parameter sets the current draw delta that is used to determine the on and off states of the motor. A value of 0 disables the switching state detection and MagicQuartz assumes that the motor is always switched on. Please refer to the section 9 on page 91 for details.

Default: *40*

Range: *0–1023*

5.4.2.23. Set MotorSwTime

The `MotorSwTime` (*Motor Switching Time*) parameter sets the delay that is used to compensate the inrush current measurement, when switching the motor on. The value is in 200 ms, i.e. a value of 5 means 1 second. Please refer to the section 9 on page 91 for details.

Default: *5*

Range: *0–255*

5.4.2.24. Set SensLowPass

The `SensLowPass` (*Sensor Low-Pass*) parameter sets the value that controls the strength of the rotation sensor's low-pass filter. More filtering is indicated by higher values. Please see the section 10 on page 93 for further information.

Default: *5*

Range: *0–255*

Note: From version 1.02 on, the upper value is 99 when using the left/right buttons in "Sensor Calibration Mode". Higher values can be set in the configuration menu.

5.4.2.25. Set QuickCalib

If set to 1, MagicQuartz performs a quicker calibration when entering "Live Quartz Mode". It then does not check for low enough delta values.

Default: *0*

Range: *0–1*

5.4.2.26. Set MaxOKDelta

Sets the maximum delta (measurement jitter) that is accepted by the automatic sensor calibration before entering "Live Quartz Mode" when `QuickCalib` is set to 0.

Default: *20*

Range: *0–255*

5.4.2.27. Set BufferSize

The `BufferSize` parameter specifies the size of the speed ring buffer, which is used in “Live Quartz Mode” to calculate the average speed of the turntable. It is advised to pick a multiple of two. Depending on how this parameter is set, the observed average speed value in “Live Quartz Mode” reacts to speed changes more promptly (smaller values for `BufferSize`) or less promptly (larger values for `BufferSize`). Please keep in mind that altering this value has no effect on the speed regulation algorithm since the AC frequency is determined independently of the displayed average speed.

Default: *40*

Range: *0–200*

5.4.2.28. Set BufferClear

`BufferClear` specifies the number of platter revolutions after which the speed buffer is cleared automatically when the platter’s speed consistently increases or decreases. In this way, you will receive more timely speed measurements in the event of significant speed variations (which should not actually occur).

Default: *8*

Range: *0–255*

5.4.2.29. Set BufferWait

`BufferWait` specifies the number of platter revolutions MagicQuartz waits before it starts buffering them in the ring buffer.

Default: *2*

Range: *0–255*

5.4.2.30. Set QrtzLatency

The `QrtzLatency` (*Quartz Latency*) parameter sets the intensity of the speed regulation in “Live Quartz Mode”. 0 means “off”, 1 “full intensity”. The regulation intensity decreases with higher values.

This is the main parameter to control the performance of the speed regulation algorithm. Please refer to section 5.3.4 on page 45 for an explanation on how to use this value.

Default: *3*

Range: *0–255*

Note: From version 1.02 on, the upper value is 99 when using the left/right buttons in “Live Quartz Mode”. Higher values can be set in the configuration menu.

5.4.2.31. Set QuartzRange

Sets the percentage of the maximum allowed AC frequency change in “Live Quartz Mode”.

Default: *10*

Range: *0–100*

5.4.2.32. Set WaitOnStart

Sets the number of platter revolutions MagicQuartz waits in “Live Quartz Mode” at start or after the motor has been switched off and on again before starting the speed regulation.

Default:	*3*
Range:	*0–255*

5.4.2.33. Set WaitOnBad

Sets the number of platter revolutions MagicQuartz waits in “Live Quartz Mode” after it has detected a “bad” measurement (see parameter `BadDelta` to adjust the threshold) before starting the speed regulation. ***Note:*** Do not set this to zero. If you do, you must set `WaitOnStart` to a value that is high enough to compensate an accelerating platter, e.g. 3.

Default:	*3*
Range:	*0–255*

5.4.2.34. Set BadDelta

Sets the delta (measurement jitter) value above which the measurements are being considered as “bad” in “Live Quartz Mode”. Zero disables the rejection. ***Note:*** Do not set this to zero. If you do, you must set `WaitOnStart` to a value that is high enough to compensate an accelerating platter, e.g. 3.

Default:	*50*
Range:	*0–255*

5.4.2.35. Set SaveHzAtSec

The `SaveHzAtSec` (*Save Hertz at Second*) parameter sets the time in seconds in "Live Quartz Mode" after which MagicQuartz saves the current AC frequency as the "starting frequency". During playback, the AC is usually reduced as needle friction decreases. If "Live Quartz Mode" is entered at a later time, the initial frequency can be restored automatically so that initial speed adjustment is not required.

If the turntable is a "drifter", meaning that it accelerates for a long time while it warms up, it is advisable to disable this function. This avoids the need for the speed control to compensate for an unnecessarily large range when the record is turned over. Zero disables this feature.

Default:	*30*
Range:	*0–255*

5.4.2.36. Set StandByOff / StandByOn

Sets the standby timer in minutes. If this time has passed, MagicQuartz will switch into standby mode. The timer is reset whenever a key is pressed, the motor is switched on or off and when platter motion is detected in "Live Quartz Mode". Zero disables the standby function.

Defaults:	*5 and 40, respectively*
Range:	*0–9999*

5.4.2.37. Set TotalHours

Sets the hours of the total uptime. This can be used to restore the time counter after a unit reset. ***Note:*** It is not possible to change minutes or seconds.

Range: *0–9999*

5.4.2.38. Set PlayHours

Sets the hours of the time in which the record player's motor was supposed to be running. This can be used to restore the time counter after a unit reset. ***Note:*** It is not possible to change minutes or seconds.

Range: *0–9999*

6. Problems and Solutions

This section covers problems that you may encounter, along with solutions.

6.1. Related to the Motor

Note: Running the "Motor Setup" function again, as described in section 4.2 on page 23, may resolve any of the motor-related problems described below.

6.1.1. General Motor Status Detection Fails

Problem: The motor status indicator in the lower right corner does not change when the turntable motor is switched on and off.

Solution: Try to manually decrease `MotorSwThrs` (*Motor Switching Threshold*) in the configuration menu (see section 5.4.2.22 on page 59).

Problem: The motor status indicator in the lower right corner blinks or behaves erratically.

Solution: Try to manually increase the value `MotorSwThrs` (see section 5.4.2.22 on page 59).

Problem: The motor status indicator disappears immediately after the motor has been switched on.

Solution: MagicQuartz probably interprets a dropping inrush current peak (see section 9 on page 91) as the motor switching off. Try to increase the value `MotorSwTime` (*Motor Switching Time*) to prolong the inrush current compensation time (see section 5.4.2.23 on page 59).

You may also use the function “Show Power Info” to determine the value `MotorSwTime`. Immediately after switching the motor on, the current peaks for a very short time and then normalizes to a stable current. Depending on the record player, this can be seen in the graphical read-out of function “Show Power Info”. The recalibration status should display “`W`” until the peak is fully gone. If it displays “`C`” during the current spike, you should increase the `MotorSwTime` value. This increases the time between the “`W`” and the “`C`”. Please refer to section 9 for more details.

6.1.2. Motor Status Detection upon Startup Fails

Problem: MagicQuartz fails to detect the correct motor status upon startup of the speedbox.

Solution: If MagicQuartz mistakenly recognizes that the motor is off, try to decrease `MotorRunThrs` (*Motor Running Threshold*). If MagicQuartz falsely detects that the motor is on, try to increase `MotorRunThrs`.

You can also use the function “Show Power Info” (section 5.4.2.3 on page 48) to manually determine the threshold parameter MotorRunThrs, above which the motor is supposed to be running. You can calculate MotorRunThrs with ($I_{MotorOn}$ + $I_{MotorOff}$) / 2.

As a last resort, you may set MotorRunThrs to 0. This causes MagicQuartz to recognize that the motor is always switched on.

6.1.3. Generated Voltages are Wrong

Problem: The generated voltages are not correct.

Solution: Correct the voltages by adjusting the parameters VoltageMtOff and VoltageMtOn, as described in section 5.4.2.14 on page 55. Please note that the voltage readings are only correct at the nominal frequency (as set with the DefaultHertz parameter). For higher frequencies, the voltage can be slightly higher and decreases for lower frequencies due to the U/f characteristic (see section 8.4 on page 89).

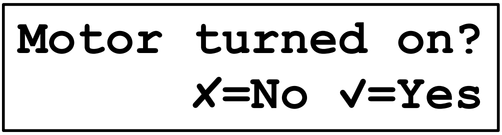
6.1.4. Overload Detection not Working

Problem: After connecting a load that exceeds the rating of the turntable, MagicQuartz does not display an overload warning.

Solution:

- □ Run the “Motor Setup” function again, as described in section 4.2 on page 23.
- □ Execute the function “Calib OvlProtThrs” (*Calibrate Overload Protection Threshold*, see section 5.4.2.7 on page 51).

6.1.5. Message: “Motor turned on?”



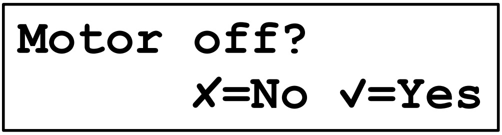
```
Motor turned on?
      ✗=No ✓=Yes
```

Problem: MagicQuartz believes that the record player’s motor is switched off and therefore does not enter the “Sensor Calibration” or “Live Quartz Mode”.

Solution:

- □ Try switching the motor off and on again. If this does not work, the current AC frequency might be too low to allow MagicQuartz to properly detect the motor status.
- □ If you’re certain that the motor is switched on, you may press the “OK” button. Be careful as this increases the voltage as specified by the `VoltageMtOn` parameter. To abort and to go back to the main menu, press the “Cancel” button.
- □ If the problem persists, check the solutions in section 6.1.1 on page 67 and section 6.1.2 on page 68.

6.1.6. Message: “Motor off?”



```
Motor off?
      ✗=No ✓=Yes
```

Problem: MagicQuartz believes that the turntable’s motor is switched on and therefore does not allow changing the parameter `VoltageMtOff` in the configuration menu.

Solution:

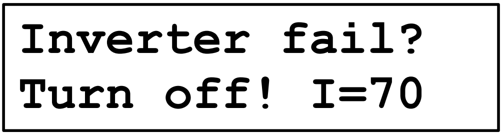
□ Try switching the motor on and off again. If this does not work, the current AC frequency might be too low to allow MagicQuartz to properly detect the motor status.

□ If you're certain that the motor is switched off, you may press the "OK" button.

□ If the problem persists, check the solutions in section 6.1.1 on page 67 and section 6.1.2 on page 68.

6.2. Related to the Inverter

6.2.1. Message "Inverter fail? Turn off!"



```
Inverter fail?
Turn off! I=70
```

Problem: The inverter, while in idle mode (i.e., no signal is currently sent to the inverter), is drawing more than twice the current as specified in the parameter `InvIdleCurr`. It is recommended to turn the speedbox off to avoid possible damage.

Solution: If the shown `InvIdleCurr` value is rather high, this may be due to a short circuit in the inverter. In this case, you may need to replace the inverter.

If the value shown is rather low and the motor is currently running, turn the motor off. Run the "Motor Setup" function again as described in section 4.2 on page 23.

If the motor is not running, switch the speedbox off and on again while keeping the “OK” button pressed. This disables APM and allows you to increase the value `InvIdleCurr` manually. Set this to a value slightly higher than the I value in the error message and restart restart the speedbox.

6.2.2. Message: “Invrt. overload! Retry?”

```
Invrt. overload!
 I=600    Retry?
```

Problem: This happens when the inverter is drawing more current than allowed by the parameter `OvlProtThrs`. If the error occurred directly after the “Motor Setup”, it is very likely that you forgot to switch the motor on again in the last step.

Solutions:

- □ Press the “OK” button to retry. MagicQuartz then attempts to restart the inverter. If this does not work and the message appears again, press the “Cancel” button to temporarily disable the output signal generation.
- □ If the problem persists, run the “Motor Setup” function as described in section 4.2 on page 23.
- □ Use the function “Calib OvlProtThrs” as described in section 5.4.2.7 on page 51.
- □ Increase the `OvlProtThrs` value or increase the value `ProtDelay` (see section 5.4.2.21 on page 58).

6.2.3. Message: “Inverter down? Retry?”

```
Inverter down?
 I=13     Retry?
```

Problem: This happens when the inverter is drawing less current than specified by the parameter `InvIdleCurr`.

Reason: The inverter has switched itself off or into protection mode because the load was too high or MagicQuartz was too fast when ramping up the voltage. This is a common problem with sensitive inverters.

Solution:

- □ Press the “OK” button to retry. MagicQuartz then attempts to restart the inverter. If this does not work and the message appears again, press the “Cancel” button to temporarily disable the output signal generation.
- □ If the problem persists, run the “Motor Setup” function as described in section 4.2 on page 23.
- □ Increase the `RampingTime` value, as described in section 5.4.2.17 on page 57.
- □ If the problem still persists, decrease the value `InvIdleCurr` or increase the value `ProtDelay` (see section 5.4.2.21 on page 58).

Reason: The inverter has switched itself off because it was running too hot.

Solution:

- □ Wait some time, then press the “OK” button to retry.
- □ Try to improve the cooling of the inverter.

6.3. Related to the Motion Sensor

Note: If there are problems with the speed measurement, check that the record player is in good mechanical condition. For example, a slipping belt or a belt rubbing against the edge of the motor pulley may cause problems similar to those described below.

6.3.1. Not Leaving Sensor Calibration

Problem: When attempting to enter "Live Quartz Mode", MagicQuartz never leaves the sensor calibration screen. The sensor contrast and/or jitter are not considered good enough by MagicQuartz to continue.

Solution:

- □ Please refer to section 5.2 on how to properly setup the sensor.
- □ If the jitter is still a problem, you may try to increase the parameter `MaxOKDelta`. If nothing else helps, you can also set `QuickCalib` to 1. Depending on how "bad" the jitter is, the speed regulation in "Live Quartz Mode" may not work satisfactory.

6.3.2. Dropped Speed Measurements in Live Quartz Mode

Problem: MagicQuartz constantly drops bad measurements (the symbol ✖ is shown in the lower right corner), pauses the speed regulation (symbol: ❚❚) and never (or rarely) displays the “Play” symbol (▶). The reason is that the jitter is too high.

Solution: Please refer to the solution of the problem above.

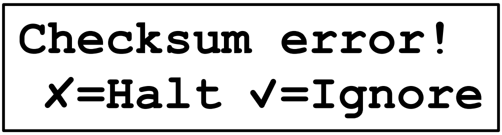
6.3.3. No Play Symbol in Live Quartz Mode

Problem: MagicQuartz constantly displays double up or down arrow symbols (⏫ and ⏬) and never (or rarely) displays the “Play” symbol (▶). The speed regulation algorithm can’t keep up with the speed changes of the turntable.

Solution: `QrtzLatency` is set too high. Reduce `QrtzLatency` to a value >= 1 by pressing the “Left” and “Right” buttons in “Live Quartz Mode”. ***Note:*** It is not possible to set `QrtzLatency` to less than 1 (0 turns the speed regulation off).

6.4. Other Problems

6.4.1. Message: “Checksum error!”

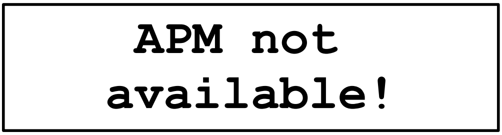


```
Checksum error!
 ✗=Halt ✓=Ignore
```

Problem: MagicQuartz has detected a problem with the data integrity of the microcontroller’s flash memory. The most likely reason is that the speedbox was turned off while it was saving data.

Solution: Since often only the values of the time counters (`PlayHours` and `TotalHours` as well as the minute counters that are not accessible to the user) have changed between the individual memory write operation, the use of the speedbox can in most cases continued by pressing the "OK" button. Note, however, that this will reset some more critical runtime values (e.g. the frequency and platter speed) to their default values for safety. The `PlayHours` and `TotalHours` parameters should also be checked in the configuration menu afterwards. If the error message occurs more frequently, the memory of the microcontroller could be defective.

6.4.2. Message: "APM not available!"



```
  APM not
available!
```

Problem: The hardware MagicQuartz is running on apparently does not support the Advanced Power Management (APM, see section 9 on page 91).

Solution: This is the expected behavior with Arduino and compatible development boards. However, if the device supports APM, check if the hardware is properly configured for it, e.g. via a jumper or a microswitch on the board. You can also check if the signal "Enable APM" is pulled to ground, as described in section 13.5 on page 105.

6.5. Emergency Access

In case of a misconfiguration of the software, it may no longer be possible to start the device correctly, which could result in the inability to correct the configuration. In order to overcome this problem, emergency access to important functions is made available during startup. To access these functions, press and hold the respective button when MagicQuartz shows the startup screen:

- □ **Left Button:** Calls the “Format Memory” function as described in section 5.4.2.8 on page 52.
- □ **Right Button:** Calls the “Motor Setup” function as described in section 4.2 on page 23.
- □ **Down Button:** Disables the output signal, thereby effectively switching the inverter off. During operation, the signal can be reactivated at any time via the “Restart Inverter” function, as described in section 5.4.2.2 on page 47.
- □ **Up Button:** Disables the Advanced Power Management (APM) in software. For an introduction to APM, see section 9 on page 91.

Please note that the emergency access is not available when using a serial connection.

7. Appendix

7.1. Graphical and Serial Symbols

When using a serial connection, no graphical symbols can be displayed, so these are replaced by text symbols.

General Symbols

Graphical:	✓	✗	⚡
Serial:	U	D	R

Symbols in “Sensor Calibration Mode”

Graphical:	↧	↥	◪	✓
Serial:	+	-	C	*

Symbols in “Live Quartz Mode”

Graphical:	▶	▲	▼	⏫	⏬	⏸	✖	■
Serial:	|	)	(	>	<	W	D	S

7.2. Revision History

Note: This history covers major and minor revisions (first and second digits), but not patches (third digit). For full details, please refer to the detailed MagicQuartz change log available on the website.

Version 0.9 (May 7, 2015)

- □ First version for the ATmega328P microcontroller.

Version 1.0 “Phoenix” (April 1, 2023)

- □ Ported to the ATmega2560 microcontroller. A single firmware binary now supports different hardware configurations, as described in section 13.1 on page 101.
- □ Many parts of the previous version fully reimplemented (“Motor Setup”, APM motor status detection) or improved (“Live Quartz Mode”, APM handling of inverter errors, flash memory handling, emergency access).
- □ New features: Voltage reduction in “Live Quartz Mode”, function to restart the inverter, license registration.
- □ Signal generation: Improved wavetables, new sine waves (`WaveForm` = 1 and 2), optional phase shift, and smooth voltage and frequency ramping.

Part II.

Theory of Operation

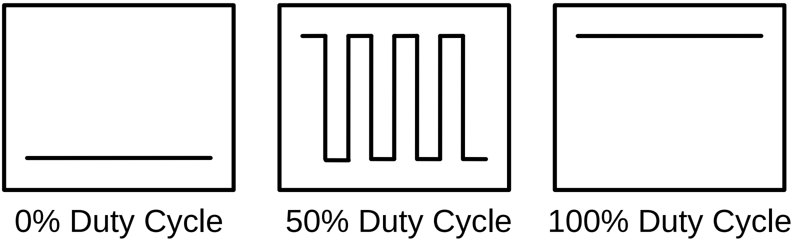
8. Signal Generation

8.1. Introduction to PWM

MagicQuartz uses a technique called pulse width modulation (PWM) to generate sinusoidal signals. PWM is a common technique for generating analogue signals from digital systems and works by rapidly switching a digital signal between its high and low states at a fixed carrier frequency.

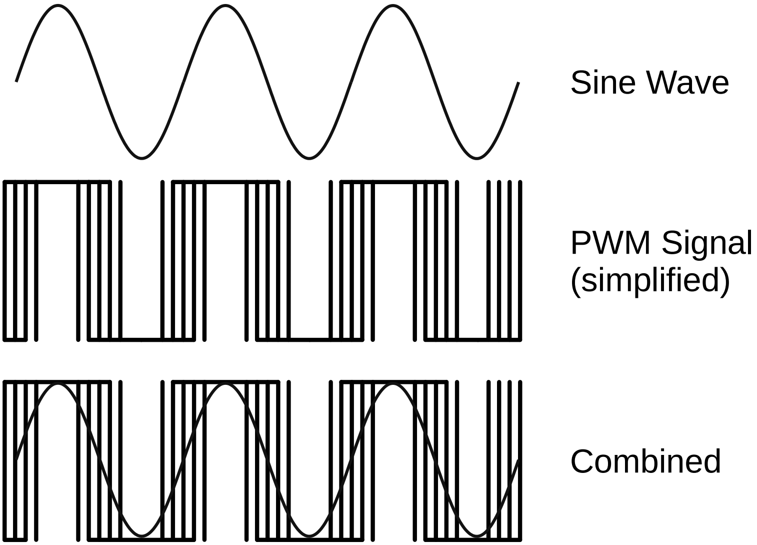
In PWM, the amount of time the digital signal remains in its high state, also known as the on-time or duty cycle, determines the corresponding analogue signal level. The duty cycle is expressed as a percentage of the total period, where 100% means the digital signal stays high all the time and 0% means it stays low all the time.

For example, if the digital signal has a maximum value of 5 V and the duty cycle is set to 50%, the digital signal will be high half the time and low half the time. As a result, the analogue signal will be at 2.5 V, which is half the maximum value. This is because the duty cycle defines the proportion of the time that the digital signal is in its high state, and therefore the proportion of the maximum value that the analogue signal represents.



However, the resulting PWM signal by nature contains high frequency components that are undesirable in many applications, such as audio or power control. To remove these high frequency components, a low pass filter is typically used to filter the PWM signal, resulting in a smooth and more stable analogue signal.

The following figure illustrates the relationship between a sinusoidal signal (top) and the corresponding PWM signal (center). The PWM signal is simplified in the sense that the duty cycle is not shown as rising and falling.



8.2. PWM Implementation in MagicQuartz

In some applications, such as an inverter driving a record player, the PWM carrier frequency can be audible in the resulting audio signal, causing unwanted noise or interference. To prevent this, MagicQuartz uses a PWM carrier frequency of 31.250 kHz, which is outside of the range of human hearing.

The generated signal is fed into the inverter, which then generates the high-voltage AC (i.e., 120 or 220 V). At full amplitude, a dynamic resolution of 8 bits (`WaveForm` 1 and 2) or 9 bits (`WaveForm` 3) with 128 discrete values (samples) per period is achieved.

Since the AC signal is reproduced at a variable sampling rate, all 128 samples of the wavetable are replayed at each AC frequency to be generated, which results in an undistorted, pure sinusoidal signal. However, since the internal timing of the microprocessor limits the variability of the playback sampling rate, it was necessary to develop a special technology, which allows finely graduated frequency changes in steps of only 0.0001 Hz. This is achieved by a clever programming of the microcontroller's hardware timers.

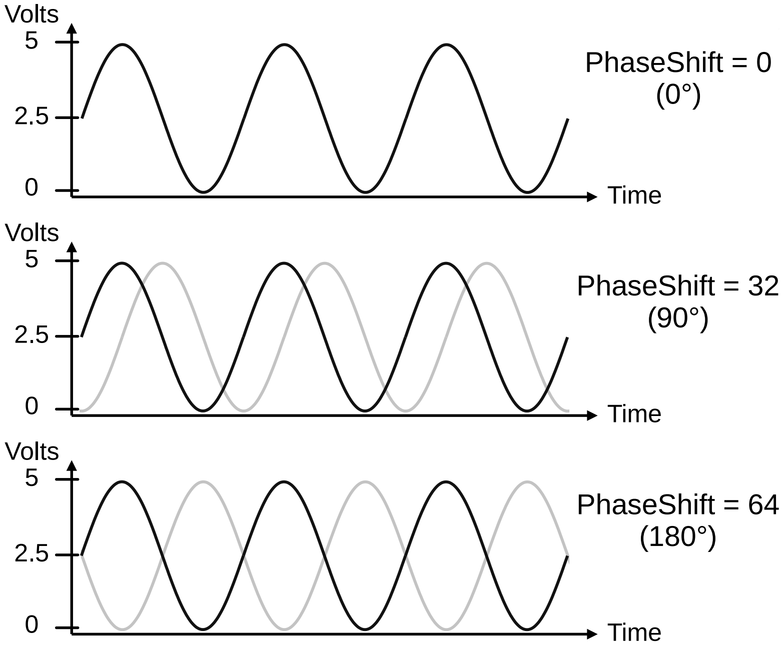
8.3. Wave Forms

MagicQuartz is capable of generating three different types of wave forms that can be set via the `WaveForm` parameter:

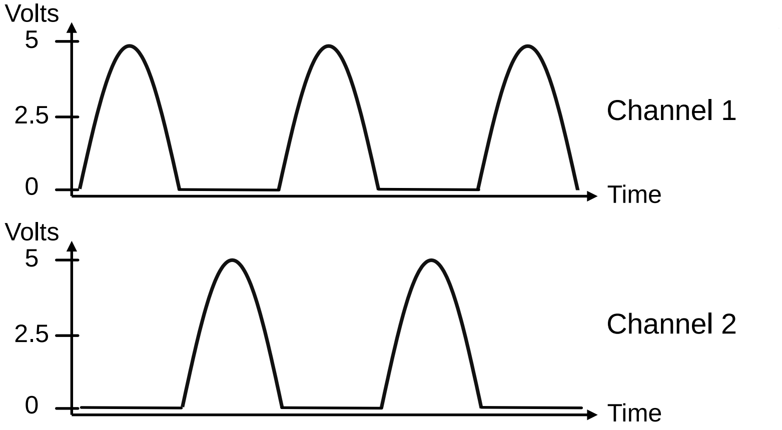
- □ **0: Disabled.** This disables the signal generation. This value is set by default to prevent possible damage to the inverter or the connected turntable.
- □ **1: Single or dual phase full sine wave with fixed 2.5 V DC offset.** Use this if the speedbox uses an analog interface to the inverter and features hardware-based pop-suppression. The generated sine is referenced to a 2.5 V offset. Since this offset is enabled shortly after switching on, this may generate a voltage pulse ("pop") in the inverter. The "DAC Active" signal (see section 13.5 on page 105) can be used to implement means on the hardware side to prevent this pulse.
- □ **2: Single or dual phase full sine wave with dynamic DC offset and software-side pop-suppression.** Use this if the speedbox uses an analog interface to the inverter and does not implement hardware-based pop-suppression.
- □ **3: Single phase half sine wave.** Use this if the speedbox uses a digital interface to the inverter.

The phase shift results from the offset in samples between the first and second channel during signal reproduction. Because the wave is stored in 128 samples, a `PhaseShift` value of 0 corresponds to 0°, 32 to 90°, 64 to 180°, and so on. For `WaveForm` = 3 it is not possible to set a phase shift because the two channels are used to reproduce the two halves of the sine wave.

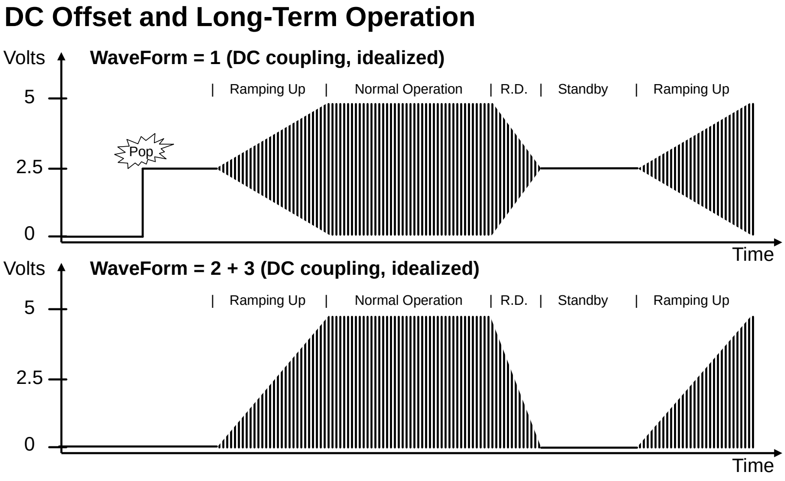
WaveForm 1 and 2 (low-pass filtered, idealized)



WaveForm 3 (low-pass filtered, idealized)



The following figure illustrates the difference between `WaveForm` 1 and 2 in longer-term operation ("R.D." = "Ramping Down"). The dashed areas are to be understood as the envelope of the sine signal. As can be seen in the upper section on the left, `WaveForm` 1 may produce a pop as soon as the generated voltage is brought to the 2.5 V offset (which is the zero level of the sine wave). This is avoided with `WaveForm` 2 by using a dynamic DC offset (the dynamic DC offset results from half voltage of the envelope).

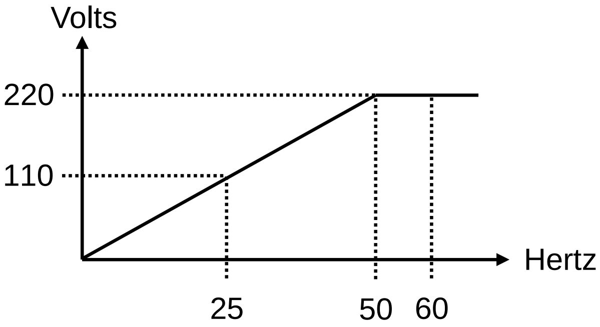


All signals generated by the microcontroller are DC signals for technical reasons. Since the signals of `WaveForm` 1 and 2 are typically brought to the inverter via an analog interface, they may have to be AC-coupled. With `WaveForm` 2, however, this has the disadvantage that drift effects may occur due to the dynamic offset of `WaveForm` 2. This is of little relevance in practice.

8.4. Voltage Regulation

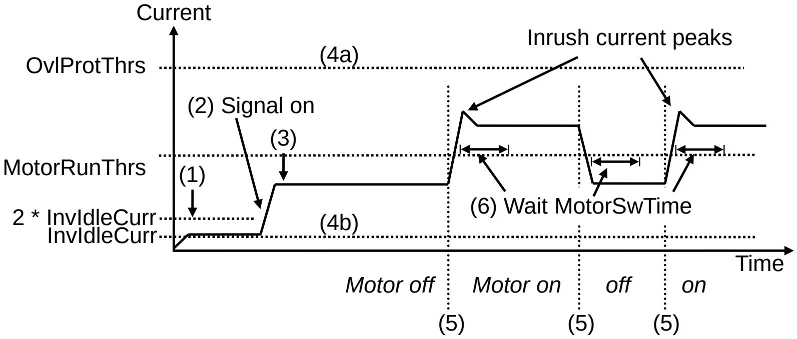
Depending on the detected motor status ("on" or "off"), the factors `VoltageMtOn` or `VoltageMtOff` are applied to the amplitude of the generated signal in order to adjust the supply voltage. Both values should be set to output exactly the rated voltage of the record player (220 or 120 V) at the rated frequency (as set in `DefaultHertz`).

MagicQuartz implements a simplified volts-per-frequency (V/f) characteristic. It is simplified in that, although the voltage decreases linearly with decreasing frequency, it does not increase when the generated frequency exceeds the rated frequency of the motor. This is done for safety reasons and to avoid damaging other electronic components in the turntable. The voltage correction factor results from the ratio of the generated frequency and the rated frequency. For example, a 220 V / 50 Hz turntable motor, when running at 25 Hz, is driven with 110 V. At 60 Hz, however, the voltage remains at 220 V:



9. Advanced Power Management

An optional feature called “Advanced Power Management” (APM), which is able to constantly monitor the inverter current, can be used to automatically detect the motor status and to intercept possible inverter fault conditions.



After switching the device on (1), the current (I) drawn by the inverter is measured before signal generation is activated. If I > 2 * `InvIdleCurr` *(Inverter Idle Current)*, MagicQuartz displays the warning “Inverter fail? Turn off!”.

MagicQuartz then activates the signal generation (2). It is possible that the turntable motor has already been switched on before the speedbox. If I > `MotorRunThrs` (*Motor Running Threshold*), MagicQuartz assumes that the motor was already turned on (3). In the figure, this is not the case, so the motor is assumed to be off.

From now on, MagicQuartz continuously monitors the current of the inverter to check if it is overloaded or in shutdown mode:

- □ I > `OvlProtThrs` (4a): Triggers an overload warning, as described in section 6.2.2 on page 72.
- □ I < `InvIdleCurr` (4b): Triggers an inverter shutdown warning as described in section 6.2.3 on page 73.

MagicQuartz also continuously detects if the motor switches on or off. In the figure above, this happens three times (5). The way the detection works depends on whether the motor is currently running at nominal frequency (parameter `DefaultHertz`) or not. At nominal motor frequency (+/– 2 Hz), the `MotorRunThrs` value is used to determine the motor switching status, as described above. Since this approach is unreliable at other frequencies, MagicQuartz uses a relative current value (d) to detect switching events. The detection sensitivity is specified via the `MotorSwThrs` (*Motor Switching Threshold*) parameter and works as follows:

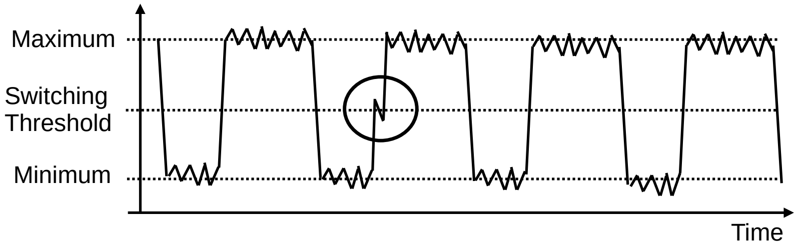
- □ d > `MotorSwThrs`: Motor has been switched on.
- □ d < –`MotorSwThrs`: Motor has been switched off.

Whenever MagicQuartz has detected a switching event, it waits the time specified in `MotorSwTime` (*Motor Switching Time*) to ignore possible inrush currents (6).

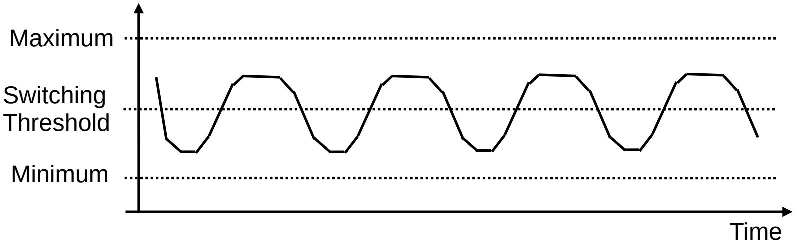
10. Rotation Sensor

The rotation sensor signal is used to determine the speed of the record player's platter. Typically, an optical sensor is used in combination with a paper sticker that is attached to the turntable's platter.

The original sensor signal is typically overlaid with a noise signal, which can cause falsely detected edges, as shown in the following figure (circle):



By applying a low pass filter, this noise can be removed. The higher the strength of low pass filter, the more noise is removed. However, the filter also smoothens the overall signal, turning it from a rectangular-like signal (as shown above) into a more sine-like signal (as shown below). This also reduces the quality of the edges and reduces the precision of the edge detection.



In MagicQuartz, such a low-pass filter can be configured by changing the parameter `SensLowPass`, where higher values mean more filtering.

The graphical display in the function “Show Sensor Info” can be used to evaluate how well the rotation signal is suited to be processed by MagicQuartz. For an optimal signal, the current sensor value (black box) flips between maximum and minimum. In case an optical sensor and a paper sticker attached to the platter is used, the bypassing sticker causes the short “low” periods. MagicQuartz is interpreting flips across the half of the detected range (dashed area), i.e. (Maximum + Minimum) / 2 as detected edges (switch threshold).

Please note that for real-world signals, the current value typically does not always flip between minimum and maximum due to the noise added to the signal (see above). This is perfectly fine as long as the “high” and “low” periods are well above and below the switching threshold.

However, if e.g. the maximum value (which in our above example is the rim of the platter and not the sticker) is not stable and goes below the switching threshold, MagicQuartz will fail to detect a “sharp” edge in the optical signal. It will detect an edge before the sticker reaches the optical sensor. This may happen if the rim of the record player’s platter does not have a constant brightness, as e.g. on polished platters.

11. Flash Memory Handling

MagicQuartz implements a comprehensive abstraction layer for the internal flash memory of the microcontroller with the aim of reducing the number of write operations to individual memory cells. It should be noted that the life expectancy of the microcontroller flash memory is by no means low, and to date there has been no experience with any memory failure. The techniques described below had been implemented solely with the aim of creating a long-lasting product:

- □ If data are to be stored, the memory cells are first read and only updated if the values differ.
- □ The firmware uses delayed writing operations. For example, when the user changes parameters on the main screen or in the menus, the data are not saved until the user has stopped entering anything or using the speedbox for 10 seconds.
- □ Similarly, runtime statistics are only saved every 5 minutes. To compensate for the fact that in average 2.5 minutes are “lost” when turning the device off, the firmware automatically adds 2.5 minutes upon startup.
- □ The data to be stored in flash memory require only a tiny fraction of the available space. To reduce the

write operations to individual memory cells, the data are shifted by one byte every 24 hours when the device enters standby mode. The initial memory position is determined randomly when the firmware is started the first time or after formatting the memory. If the device has not entered standby mode for 48 hours, the memory shift will occur the next time the device is powered up.

□ The data integrity of the flash memory is tracked via a checksum.

□ Each time data are written to the memory, an LED lights up briefly, depending on the hardware design (see section 13.5 on page 105).

Part III.

Hardware Interface

12. Microcontroller Configuration

The microcontroller must be operated with a 16 MHz clock. To increase accuracy, a quartz crystal should be used instead of a ceramic resonator. Note that the Arduino and compatible development boards usually employ a ceramic resonator. See section 14.1 on page 107 for details.

The fuse bytes should be configured as follows:

- □ **Low Fuse:** FF
- □ **High Fuse:** D8
- □ **Extended Fuse:** FD

If these have not yet been changed manually, they should be configured this way on any new Arduino or compatible development board.

13. Pin Utilization

This section describes the functions of all microcontroller pins used by MagicQuartz. In the following, the mentioned pins refer to the ATmega2560 microcontroller. The corresponding Arduino pins are given in brackets. For example, “PF0 (A0)” refers to the ATmega2560 pin “PF0” and Arduino pin “A0”.

The following documentation does not describe the common ATmega2560-related circuit sections, such as the power supply, reset line, connection of the crystal oscillator, serial port and others. Please refer to the official documentation from Microchip. The schematics of Arduino boards are also a good starting point for own developments.

Important: For all electrical parameters of the ATmega2560 microcontroller, which is also used on the compatible development boards, please refer to Microchip’s official documentation. Note that the ATmega2560 is a 5 V microcontroller and for input signals, the voltage must not exceed 5 V.

13.1. I/O Modes

MagicQuartz was developed with the goal of high adaptability to different hardware designs. For this reason, the firmware (in the form of a single binary that is flashed to the microcontroller) supports three different I/O modes:

- □ **I/O Mode 1 (Default):** The pin use of the microcontroller has been optimized for the “MEGA 2560 PRO” board. By using only the outer pins of the board, it can be easily integrated into own hardware designs (e.g., as daughterboard) or used on a breadboard.
- □ **I/O Mode 2 (LCD keypad shield):** The pin use of the microcontroller has been optimized to support the typical LCD keypad shields on an Arduino board (such as the Arduino Mega 2560 Rev3) while ensuring that the LCD keypad shield does not block or cover the other required pins.
- □ **I/O Mode 3 (Serial):** There are no buttons or a display. A serial communication at 57,600 baud with ANSI escape codes is used to interface with MagicQuartz. This allows executing the MagicQuartz firmware directly on an Arduino board (such as the Arduino Mega 2560 Rev3) without any additional hardware required. Using the serial mode is described in section 4.1.5 on page 19.

MagicQuartz detects the mode by itself upon startup based on the following conditions:

- □ If PA3 (D25) is pulled to ground: *I/O Mode 1*
- □ If PF0 (A0) $>$ 2.5 V: *I/O Mode 2*
- □ If none of the above apply: *I/O Mode 3*

13.2. Buttons

In I/O Mode 1, the buttons should be connected so that they pull the corresponding pin of the microcontroller to ground when pressed.

In I/O Mode 2, an analog voltage is measured via the ADC. The voltage can be generated in the circuit, for example, via a resistor network (please refer to the schematic of a typical LCD keypad shield for an example).

All buttons are debounced in software.

Button	Direction	I/O Mode 1	I/O Mode 2
Left	Input	PA1 (D23)	PF0 (A0) < 0.25 V
Right	Input	PD0 (D21)	PF0 (A0) < 0.95 V
Down	Input	PD2 (D19)	PF0 (A0) < 1.86 V
Up	Input	PH0 (D17)	PF0 (A0) < 2.70 V
Select	Input	n/a	PF0 (A0) < 3.86 V

13.3. Display

MagicQuartz can drive a 2 lines by 16 character display that uses an HD44780 or compatible controller (e.g., KS0070, ST7066U) in 4 bit mode, which only uses the DB4-DB7 data lines. The lower data lines of the display are to be left open.

Please refer to the data sheet of the display regarding contrast control and backlight.

Display	Direction	I/O Mode 1	I/O Mode 2
DB7 (Pin 14)	Output	PJ0 (D15)	PH4 (D7)
DB6 (Pin 13)	Output	PB5 (D11)	PH3 (D6)
DB5 (Pin 12)	Output	PH6 (D9)	PE3 (D5)
DB4 (Pin 11)	Output	PH4 (D7)	PG5 (D4)
E (Pin 6)	Output	PE3 (D5)	PH6 (D9)
R/S (Pin 4)	Output	PE5 (D3)	PH4 (D8)

13.4. PWM and Clock Signals

Signal	Direction	Any I/O Mode
PWM Channel 1	Output	PL3 (D46)
PWM Channel 2	Output	PL5 (D44)
50 Hz Reference	Output	PG1 (D40)
Regulated Frequency	Output	PL7 (D42)
(Reserved)	Output	PD7 (D38)

The PWM channels 1 and 2 are used to generate the sine waves used to drive the inverter as described in 8.3 on page 86.

The “50 Hz Reference” signal is a rectangular fixed 50 Hz reference signal. The “Regulated Frequency” signal is a rectangular signal with the same frequency as the regulated target frequency (as for PWM channels 1 and 2).

13.5. Other Signals

Signal	Direction	Any I/O Mode
Platter Rotation Sensor	Input	PF4 (A4)
Enable APM	Input	PA5 (D27)
APM Current Sense	Input	PF2 (A2)
Enable I/O Mode 1	Input	PA3 (D25)
LED	Output	PB7 (D13)
Not in Standby	Output	PB4 (D10)
DAC Active	Output	PC1 (D36)

The “Platter Rotation Sensor” input expects an analog voltage between 0-5 V and can be used to measure the rotational speed of the turntable’s platter, as described in section 10 on page 93.

“Enable APM”, when pulled to ground, enables the “Advanced Power Management” (APM) feature. The “APM Current Sense” input expects an analog voltage between 0–5 V and is used to measure the current draw of the inverter. Please refer to section 9 on page 91 and section 8.4 on page 89 for details.

“Enable I/O Mode 1”, when pulled to ground, enables the I/O Mode 1 (see section 13.1 on page 101).

The “LED” signal is high to turn an LED on. It is used to rapidly flash the default Arduino LED upon startup to indicate that the MagicQuartz firmware has been flashed successfully (see section 4.1.5 on page 19 for details). During normal operation, it will also turn high when data is written to the flash memory of the microcontroller, as described in section 4.1.5 on page 19.

The “Not in Standby” signal is high during normal operation and low when MagicQuartz is in standby mode. This signal can be used to turn the LCD backlight on and off.

The “DAC Active” signal is high 100 ms before MagicQuartz starts generating the sine signals for the inverter (PWM channels 1 and 2) and is low 100 ms after it has stopped generating these signals. The “DAC Active” signal can be used in more complex inverter designs to mute or completely disable the inverter and implement hardware-based pop suppression.

14. Compatible Hardware

14.1. Development Boards

MagicQuartz should work fine with any ATmega2560-based development board. I have successfully tested the **Arduino Mega 2560 Rev3**, the **ELEGOO Mega-R3**, and another older, unbranded clone. These boards are great for prototyping when used in combination with an LCD keypad shield (see below). However, as they use ceramic resonators to generate the clock signal, they typically provide less timing accuracy.

In contrast, the **Mega 2560 PRO (Embed)**, while it cannot be used with an LCD keypad shield easily, is the only development board that employs a real quartz oscillator. To my knowledge, this board has been developed by a company called RobotDyn. Unfortunately, it no longer seems to be available, but there are many replicas available from other companies. I have successfully tested the **ARD MEGA2560PRO** by Joy-IT (SIMAC Electronics GmbH) and another unbranded **MEGA 2560 PRO** board. ***Note:*** Do not confuse this board with the very similar “MINI” variant, which does not have a USB socket.

14.2. LCD Keypad Shields

MagicQuartz should work fine with the LCD keypad shields that are available for Arduino and compatible development boards. I have successfully tested the **LCD keypad shield** by D1 ROBOT and an older **SainSmart LCD keypad shield V1.0**.