Software Defined Radio

SDR Hands-on Book

Burkhard Kainka

elektor

LEARN DESIGN SHARE

SDR Hands-on Book

Software Defined Radio

Burkhard Kainka

elektor

LEARN DESIGN SHARE

This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
78 York Street, London W1H 1DP, UK
Phone: (+44) (0)20 7692 8344

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 978-1-907920-76-9

Prepress Production: D-Vision, Julian van den Berg
First published in the United Kingdom 2019
Printed in the Netherlands by Wilco

Foreword **9**

Chapter 1 • The Software Defined Radio **10**

1.1 Arduino SDR Shield 10

1.2 Getting started with G8JCFSDR 15

1.3 Tuning software. 18

1.4 Installing SDRsharp 20

Chapter 2 • Practical Shortwave reception **24**

2.1 Shortwave broadcasting 24

2.2 Stations and transmission times 26

2.3 Bandwidth and sidebands 27

2.4 DRM reception. 29

2.5 Morse telegraphy. 30

2.6 Single sideband signals 30

Chapter 3 • Signal-to-noise ratio and interference signals **32**

3.1 Interference on the audio line 32

3.2 Overtone mixing 35

3.3 FM signals 36

3.4 Arduino sources of interference 37

3.5 Domestic interference signals 38

3.6 Overloading and intermodulation. 39

Chapter 4 • Antennas **41**

4.1 Wire antennas. 41

4.2 Ground separation 42

4.3 Dipole antenna 44

4.4 Sheath current Coax braid trap 45

4.5 Loop antennas 49

4.6 Antenna comparison 50

4.7 Antenna matching and preselection 52

4.8 Active antennas. 55

Chapter 5 • Tuning Software **57**

5.1 Control of the PLL device 57

5.2 Calibration 59

5.3 Universal VFO firmware 63
5.4 PC Software 65
Chapter 6 • Digital Operating Modes 67
6.1 Decoding FT8 67
6.2 Decoding WSPR. 69
6.3 Virtual audio cables 73
6.4 Digital operating modes with fldigi. 74
Chapter 7 • SDR Measurement Technology 79
7.1 Auxiliary carriers 79
7.2 Calibrating the receive levels 79
7.3 Measurements on microcontroller systems 81
7.4 Display of band occupancy 81
7.5 Four-pole measurements 84
7.6 Two-pole measurements. 87
7.7 Standing Wave Ratio measuring bridge 90
Chapter 8 • Standalone Receiver 94
8.1 Tuning without PC 94
8.2 Direct mixer 96
8.3 The IQ detector 98
8.4 CW filter. 100
Chapter 9 • WSPR Transmitter. 102
9.1 JTEncode 102
9.2 Frequency calibration 105
9.3 WSPR frequencies and start 108
9.4 200 mW transmitter amplifier 110
9.5 Antenna matching 113
9.6 CAT control 115
Chapter 10 • A QRP Transceiver. 117
10.1 Morse keying 117
10.2 Transmitter tuning 120
10.3 Automatic CQ call 123
10.4 WSPR transmit function 125

10.5 The transmitter output stage. 126
10.6 Operating elements 131

Foreword

The Elektor SDR Shield is a versatile shortwave receiver up to 30 MHz. Together with an Arduino and the appropriate software, not only radio stations can be received, but also Morse signals, SSB stations and digital signals. Why use shortwave, when nowadays higher and higher frequencies are used? Because direct connections over thousands of kilometers are possible on shortwave, even with very low transmission power.

Shortwave has always fascinated me. After them first radio tinkering I quickly turned to amateur radio. My ambition was to build as many devices as possible myself. This was more or less successful, so that I could actively participate in amateur radio for many years. Then I felt like so many: the strain in occupation and an unfavorable housing situation led to the fact that the hobby was neglected.

The development of software-defined radio has contributed to a return to shortwave. After a long time, I was able to listen to amateur radio again and gain new experiences. I am now trying to get a fresh start in amateur radio. Many problems must be solved. The right equipment must be produced, optimal antennas must be built, and old knowledge has to be refreshed.

Especially the Elektor SDR Shield has proven to be a great help. Antennas could be tested with it and new operating modes were trialed. In addition, the SDR Shield was used as a versatile measuring device to measure noise margins or antenna characteristics. Gradually the SDR was extended to a complete CW transceiver. Now I can actively participate in amateur radio operating again.

In this book I have compiled many things that were useful to me along the way, and probably to many others. Many formerly active radio amateurs have a long break behind them. And many newcomers fight with the same pitfalls. Whether it is amateur radio or DX radio reception - the tasks are similar. I therefore hope that with this book I can contribute to facilitating the way into shortwave.

I wish you good reception at all times!

Yours, Burkhard Kainka, DK7JD

Chapter 1 • The Software Defined Radio

The frontend of a simple Software Defined Radio (SDR) consists of two direct mixers, which mix down the same antenna signal with a phase difference of 90 degrees into the AF range (Figure 1.1). The I and Q signals are then coupled to the PC via the stereo input, digitized by the sound card, with a suitable SDR software. Almost all essential characteristics of the receiver are determined by the software. The hardware, on the other hand, remains relatively simple.

1.1 Arduino SDR Shield

Elektor already built a first software defined radio with USB interface in 2007, at that time with a conventional circuit board and only a few SMD components. In the meantime, there have been some musings about a new edition from time to time. But the PLL chip used at that time is no longer manufactured. Thus, a new solution had to be found. It came in the form of the Silicon Labs SI5351 PLL chip: a CMOS clock generator from 8 kHz to 160 MHz with I^2C bus.

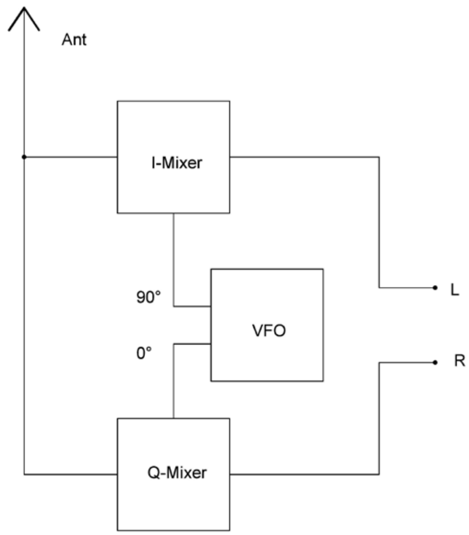

Figure 1.1: Block diagram of an SDR frontend

The first attempts were made with an Adafruit breakout board. The existing software examples were written for the Arduino, which is why the first attempts were made with the Arduino. The new VFO (see Figure 1.1) was simply connected to the existing SDR board and could demonstrate its suitability. And so, the idea was born: why not build the whole receiver as an Arduino-Shield? This way, the power supply is already clarified, and the USB interface is available.

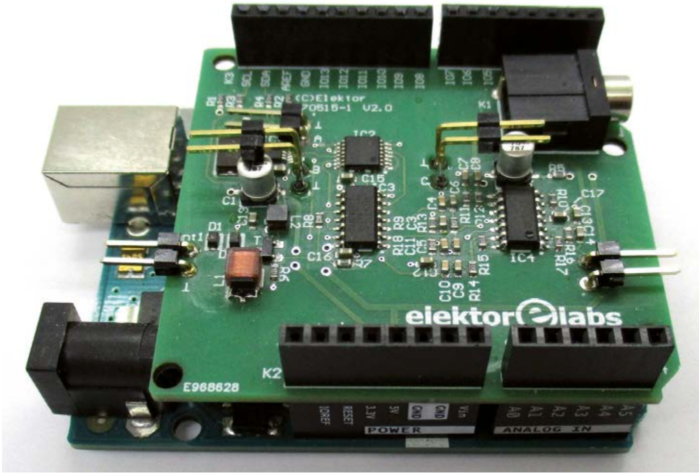

Figure 1.2: The SDR shield on an Arduino Uno

The receiver is now supplied as a fully assembled circuit board with supplied contact rows. These have to be soldered in manually so that the SDR board can be mounted on an Arduino. Next, some software has to be installed, which can be found in the software archive of the book on the Elektor web page and on the author's homepage. You can then establish a connection to the PC sound card, connect an antenna, and you're ready to go.

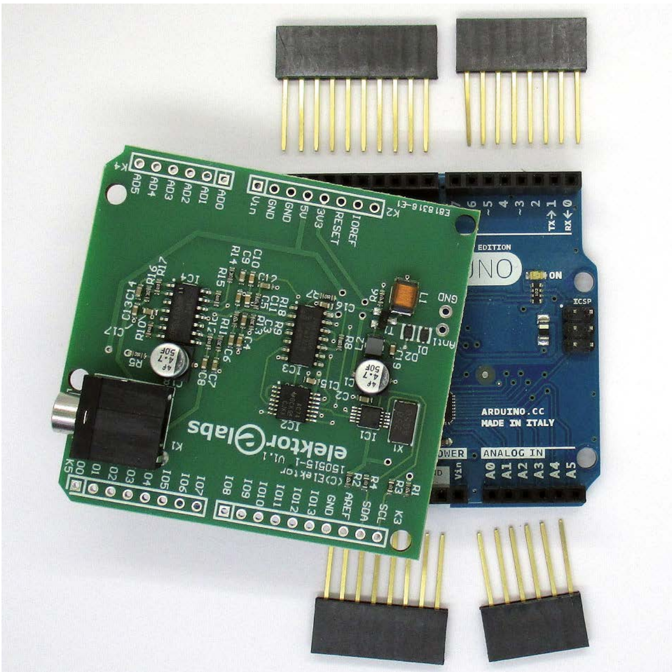

Figure 1.3: The shield and its connectors

The Arduino itself has not much to do: it simply receives serially the desired frequency from the PC and then adjusts the VFO accordingly. So, you even have a real chance to build a standalone receiver (see Chapter 8). The Arduino alone can also handle the tuning. This opens up unlimited possibilities, especially because the Arduino is widespread, and many can program it.

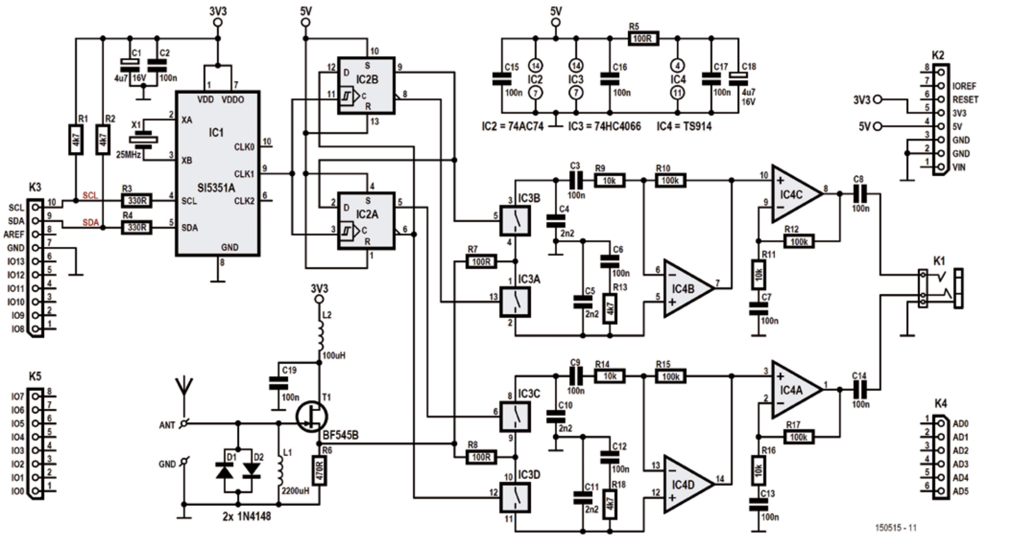

Figure 1.4: Circuit diagram of the board

A look at the circuit diagram (Figure 1.4) shows the individual modules. The SI5351 supplies the oscillator signal at four times the receiving frequency to the divider 74AC74 (IC2B). This divides the frequency by 4 and delivers the 90-degree phase-shifted signals to the 74HC4066 (IC3) mixer. This analog switch is wired like a changeover switch and sends the RF signal alternately to the inverting and the non-inverting input of the operational amplifier TS914 (IC4). This mixes the signal down into the AF range. After a small amount of filtering and amplification, the signal reaches the audio output. The RF input stage forms a source follower with the JFET BF545B, the SMD equivalent of the BF245B.

The input is broadband and is protected against overloading by two limiter diodes. This is enough for shortwave reception with a wire antenna. The overvoltage protection is included to ensure that input stages are not damaged during a thunderstorm. For critical tasks you can use external filters and preamplifiers.

The latest version V2_0 is technically identical but has additional connection points for additional PLL outputs and DC-coupled signal outputs (Figure 1.5). This extension facilitates experimental handling of the shield and simplifies external extensions for measuring instruments or shortwave transceivers.

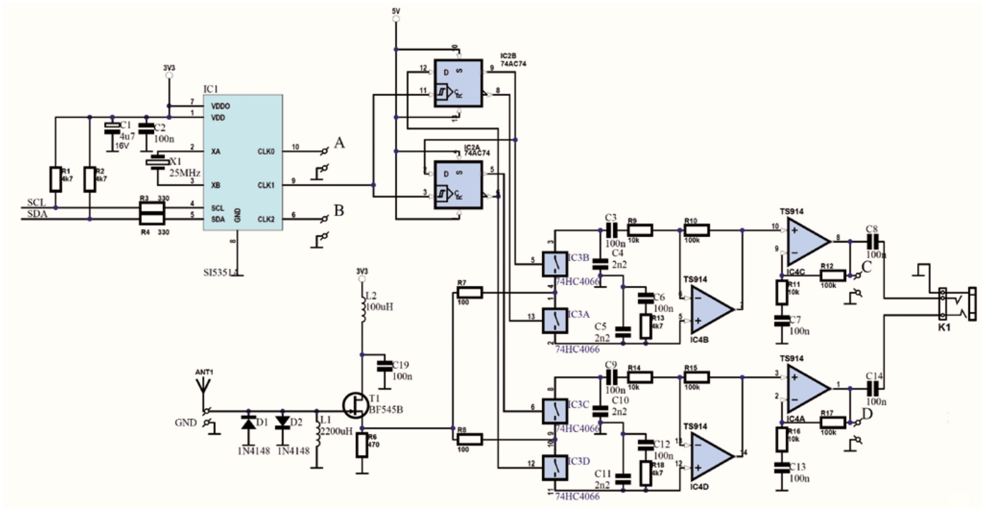

Figure 1.5: New connections in the old schematic

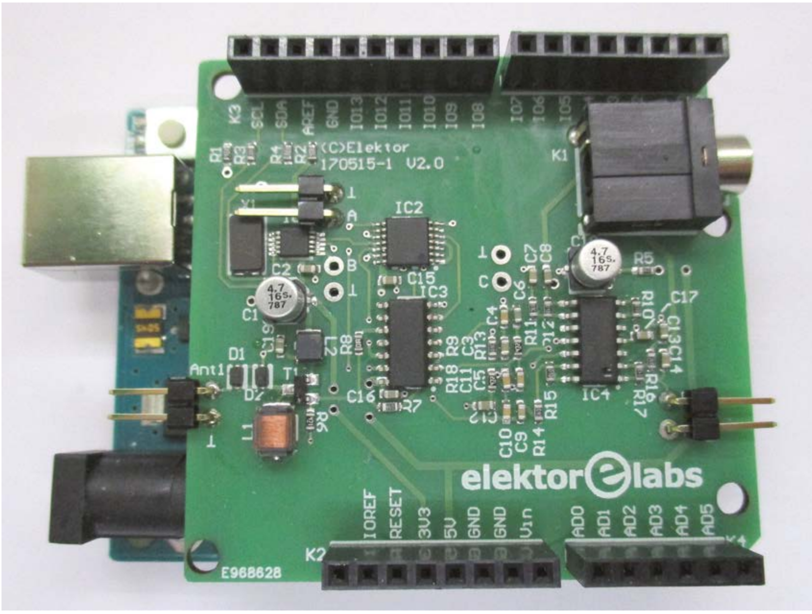

Figure 1.6: Additional connection points

To use the receiver, you need a USB connection to the Arduino, an audio cable to the audio input of the PC sound card, and a suitable antenna. In addition, an Arduino sketch must be loaded to set the VFO chip to the desired frequency.

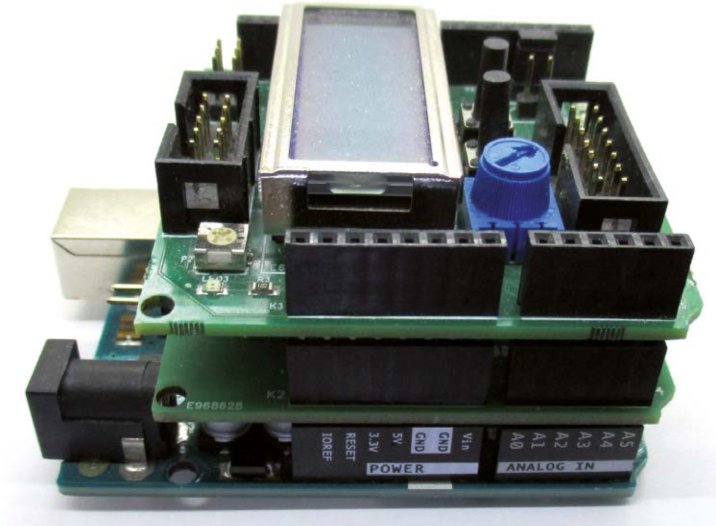

Figure 1.7: The SDR Shield stacked with Arduino Uno and the LCD-Shield

The shield is designed so that it can be used together with the Elektor LCD-Shield. It can be used to display the current frequency. It is also suitable for use as a measuring instrument or for standalone applications.

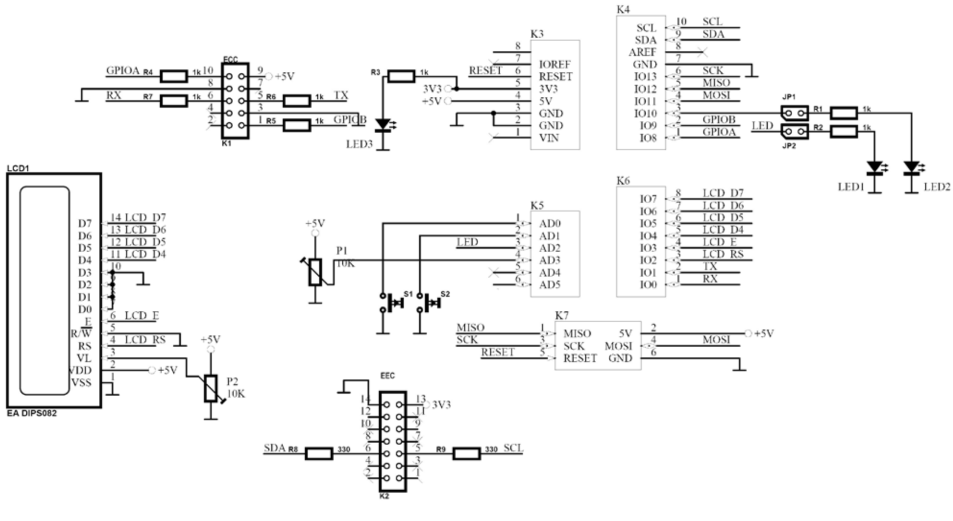

Figure 1.8: Circuit diagram of the LCD shield

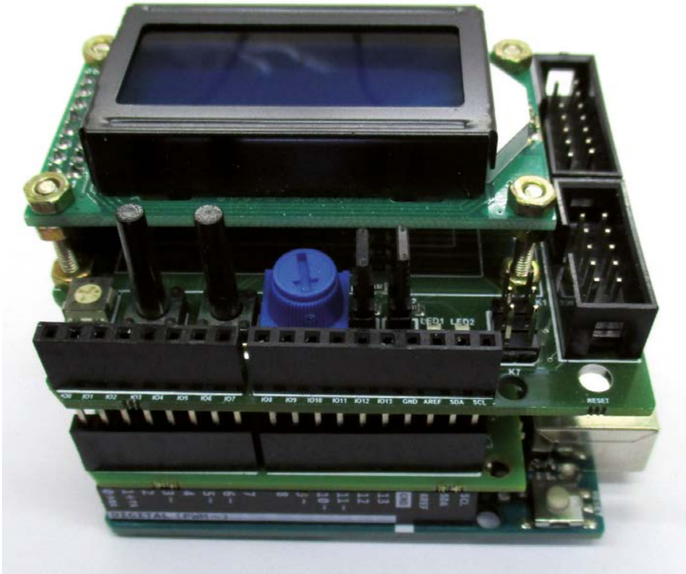

Figure 1.9: The newer Display Shield

The LCD-Shield is available in a newer, compatible version with a different LCD and a different viewing direction (Figure 1.9). It can be found under the name Arduino Experimenting-Shield 2.0 at Elektor.

1.2 Getting started with G8JCFSDR

The easiest way to get started is with the SDR program (Figure 1.10) by Peter Carnegie http://www.g8jcf.uk. It was already introduced in Elektor in 2007 and has been constantly expanded ever since. So now a Hz-exact tuning has become possible, and even an individual calibration of the VFO becomes child's play. The program automatically loads the appropriate firmware into the Arduino, so you don't even have to work with the Arduino IDE. Plug it in, switch it on, and you're done.

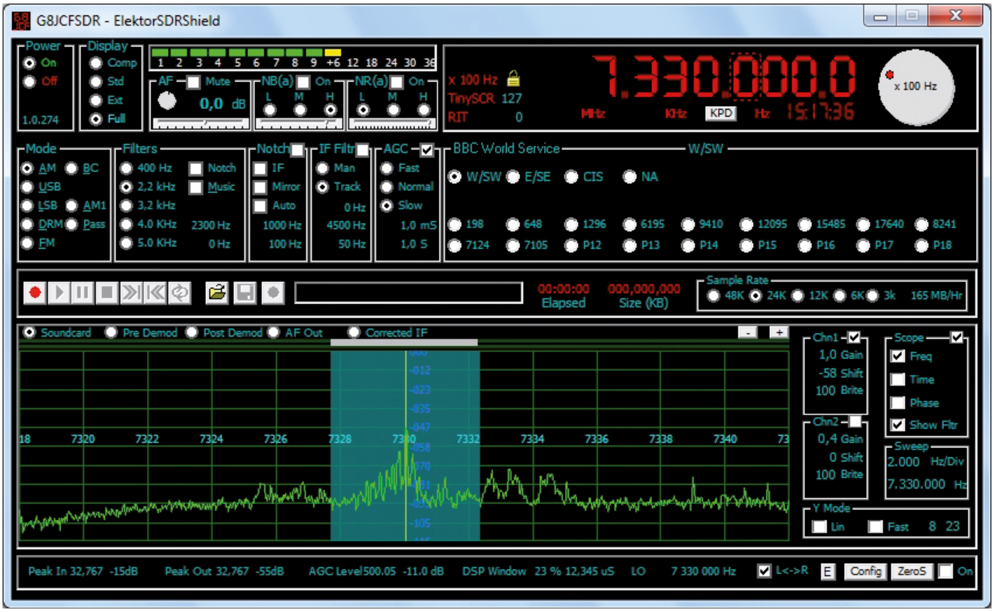

Figure 1.10: Reception of a radio station

The SDR is still "Off" the first time it is switched on. First you have to select the desired hardware (Figure 1.11). Furthermore, you have to set the user interface (e.g. COM2) under Serial I/F. The first time the program is switched on, it determines whether the required firmware is already available. If not, a window opens with an automatic upload (Figure 1.12). This is a great help for users who usually have little to do with the Arduino. No more problems with compiling, software versions and other difficulties.

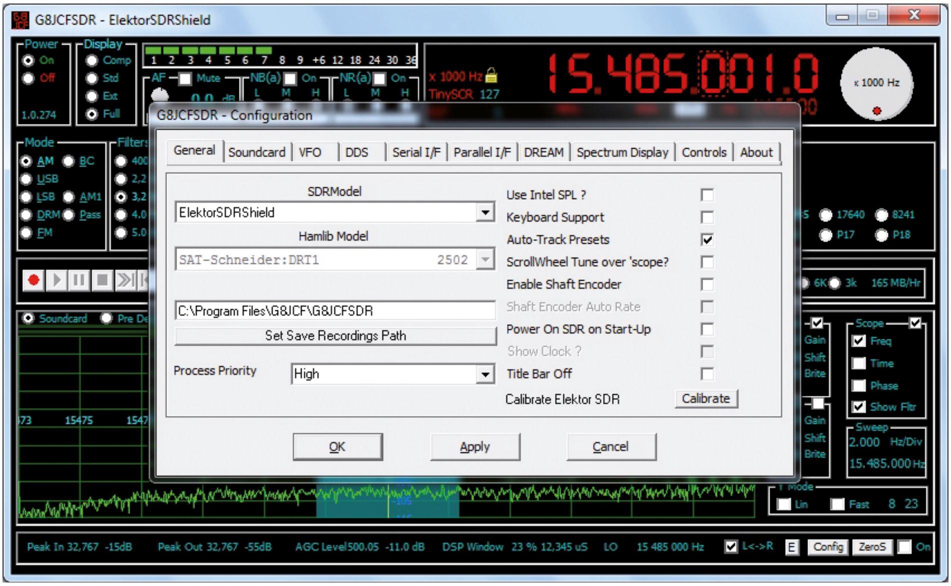

Figure 1.11: Selection of the receiver

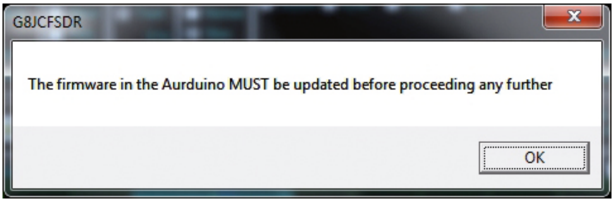

Figure 1.12: Confirmation for upload

For some applications, the basic accuracy of the VFO is not sufficient because the 25 MHz oscillator may have deviations of a few kHz. Calibration helps here. You can set a station with known frequency, and then click on Calibrate to open another menu (Figure 1.13). Then you adjust the transmitter as exactly as possible and click on Apply. This calibrates the VFO. For an optimal calibration you should set the USB position to zero beat with a carrier of a well-known radio transmitter, whereby the oscilloscope display of the NF output can help (AF OUT, Scope>Time). Depending on the history of the Arduino used, there may be a completely wrong calibration at first. In that case, a click on Reset is enough to get to the basic setting where you can expect an error of a few kHz.

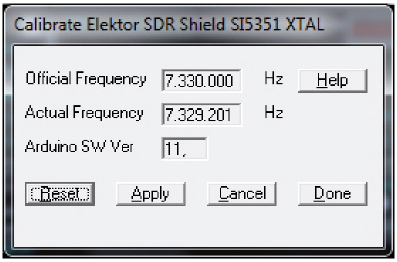

Figure 1.13: Calibration of the VFO

The receiver is now ready for use. All signals from AM broadcast through CW and SSB can be observed. Digital operating modes can also be decoded if the appropriate additional software is used.

1.3 Tuning software

Most SDR programs, unlike G8JCFSD, have no direct frequency control for the Elektor SDR Shield. Therefore, one must start a program on the PC beside the SDR software for the tuning. Usually a coarse frequency is given, while the fine tuning is done by shifting in the SDR spectrum.

In addition, a suitable firmware must be uploaded to the Arduino. Load the sketch si5351vfo2_1.ino into the Arduino IDE. The program initializes the PLL chip and sets a VFO frequency of 10100 kHz (the beginning of the 30m amateur radio band). Then it waits for commands to tune to other frequencies.

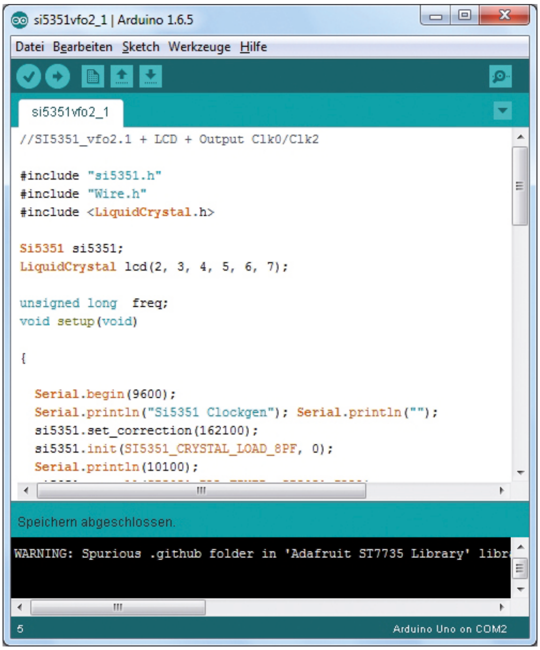

Figure 1.14: Loading the firmware

Start the serial monitor and set a transfer rate of 9600 baud. With the start of the serial monitor a reset of the Arduino is executed and the sketch restarts. You will receive a switch-on message and a display of the current frequency in kHz.

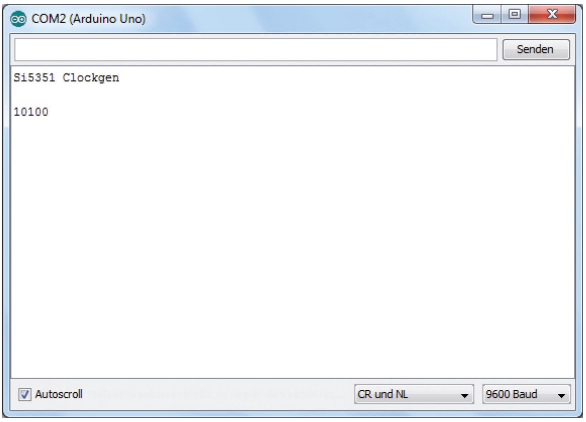

Figure 1.15: Start message in the serial monitor

Then enter a frequency in kHz. To set the receiver VFO to 7 MHz, send F7000. The Arduino reports the set frequency to the terminal. The new frequency 7000 kHz is now set.

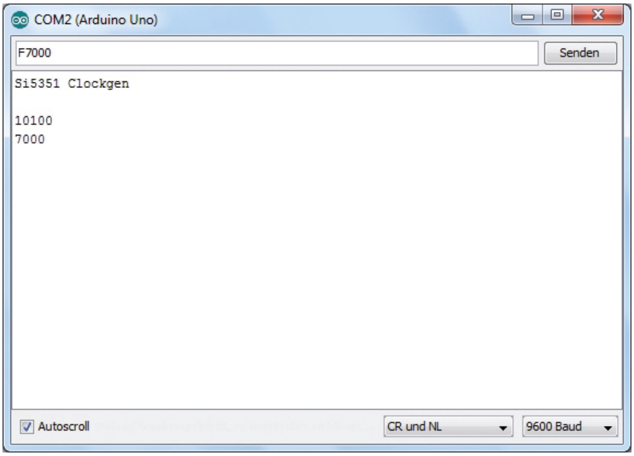

Figure 1.16: Setting to 7 MHz

A suitable PC program is used to conveniently set the desired frequency. The default program for coarse tuning is called SDRshield2_0.exe and was written in VB6. The set frequency is sent to the Arduino via a virtual serial interface. For this purpose, the COM interface used (e.g. COM2) must only be specified at the first start and opened with "Open COM". The interface used is stored in an INI file and used again the next time it is started.

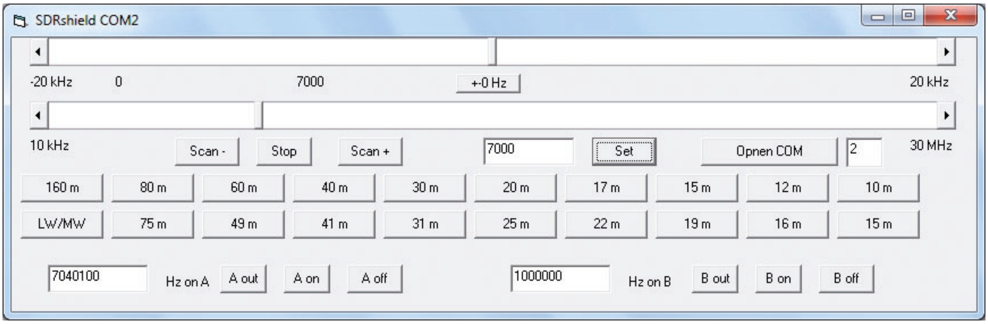

Figure 1.17: The VB program SDRshield

The lower slider allows you to adjust the VFO frequency in 1-kHz increments or in large steps of 25 kHz. In addition, a desired frequency can be entered directly and transmitted with Set. The upper slider is used for fine tuning with a resolution of 20 Hz.

The band selection buttons set the frequency to the beginning of the most important amateur radio bands and broadcast bands. This gives you the quickest overview of the current belt allocation.

The additional outputs A and B can be tuned and switched on at the lower edge. They serve special tasks, which will be explained in more detail below.

1.4 Installing SDRsharp

SDRSharp (SDR#) is software from the company Airspy (https://airspy.com). The Airspy hardware is an SDR frontend for the frequency range 24 MHz to 1800 MHz. The software also supports other receivers, though, and, above all, any device with access via the sound card. First download the Windows SDR Software Package at https://airspy.com/download.

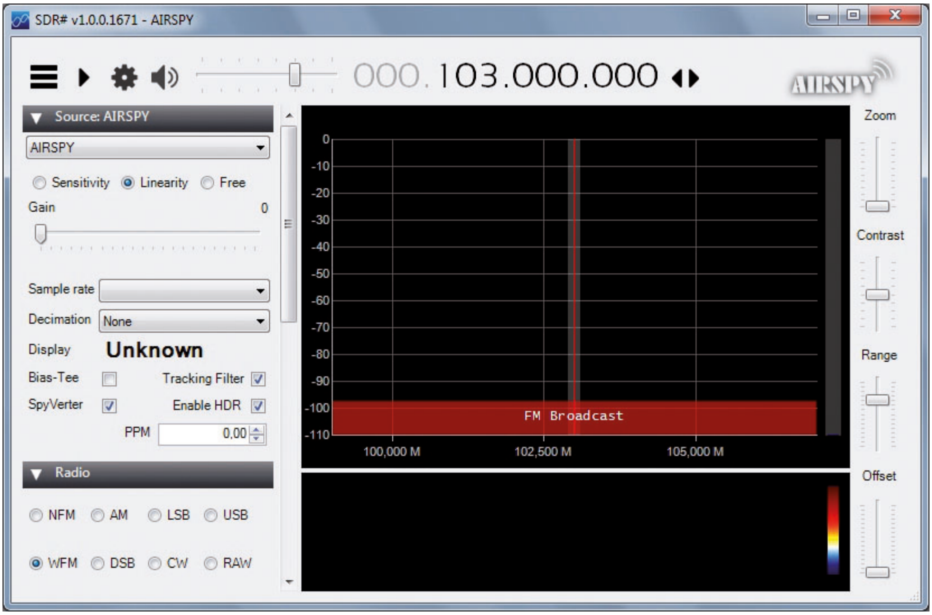

Figure 1.18: SDR# after the first start

At the first start the Airspy hardware is still set. In the Source menu, select "IQ from Sound Card" as the new source.

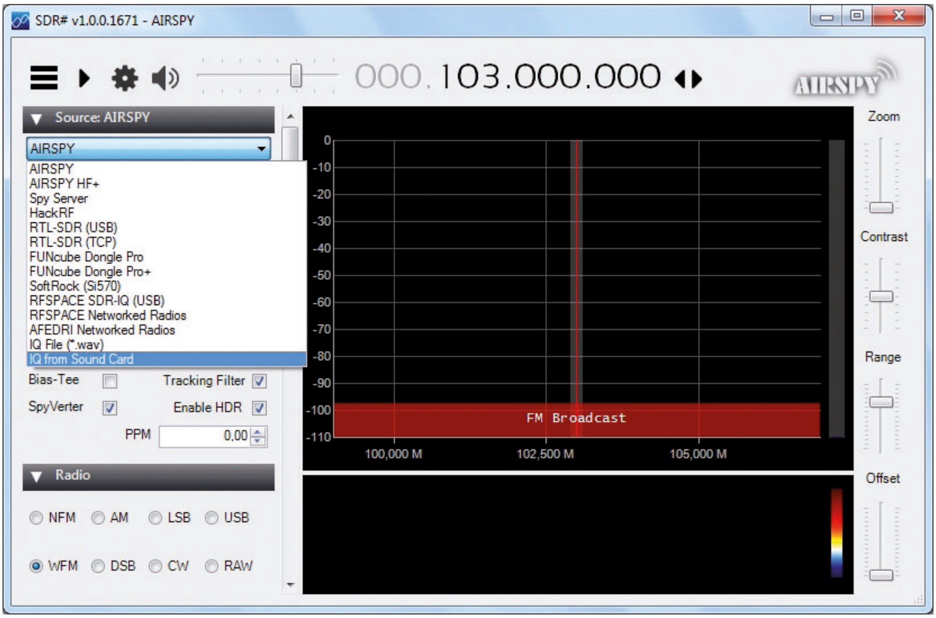

Figure 1.19: Hardware selection

In addition, the sound card and the audio input used must then be selected under the Audio menu. This can be the line-in input or the microphone input, provided it is a stereo input.

The sample rate can also be selected here. At a sampling rate of 48000 samples per second, a frequency of up to 24 kHz is transmitted. The spectrum then shows a range from -24 kHz to +24 kHz, related to the set receiver frequency, i.e. a total range of 48 kHz. Depending on the sound card, 96 kHz or even 192 kHz can be selected.

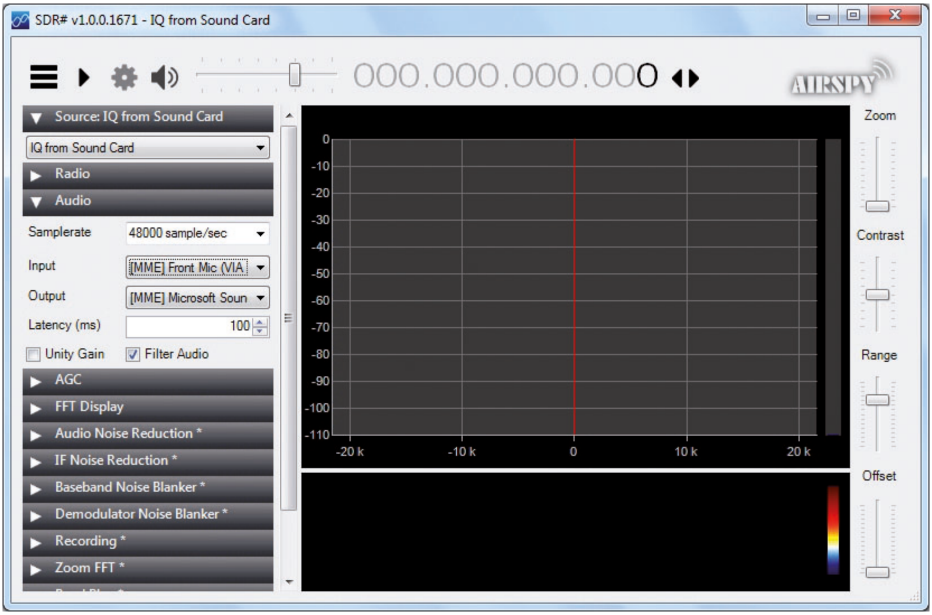

Figure 1.20: Settings for the sound card

Then start the software with the Start button in the upper line, to the left of the volume control. Normally you will already see a background noise in the spectrum. Plug in an audio cable. Whether you have found the right entrance can easily be determined by touching the open plug with your finger. This is because a hum signal with harmonics and interference signals is coupled in, which becomes clearly visible in the spectrum.

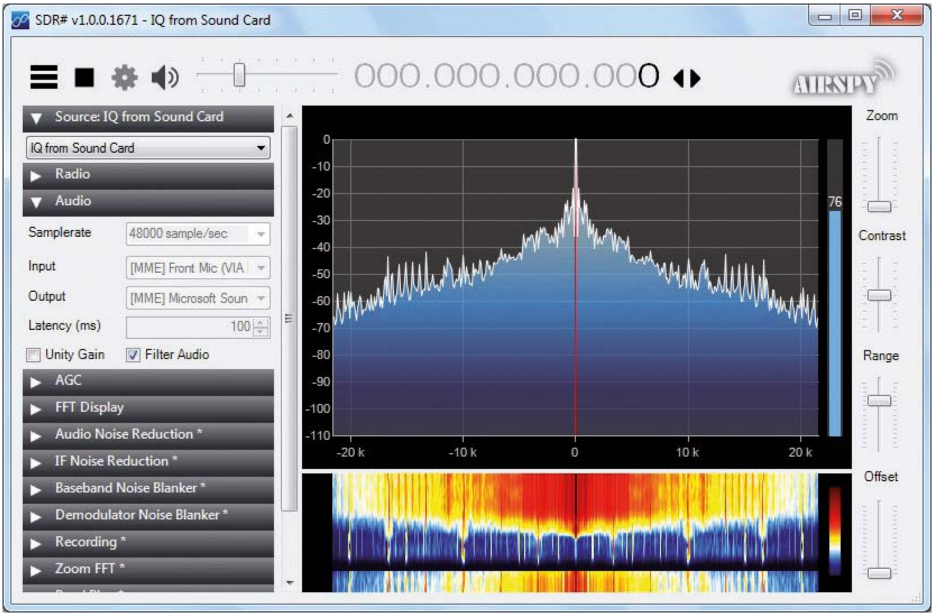

Figure 1.21: Finger test of the audio input

This completes all important preparations so that the receiver software is ready for use. Plug the audio cable into the output of the SDR shield and connect an antenna.

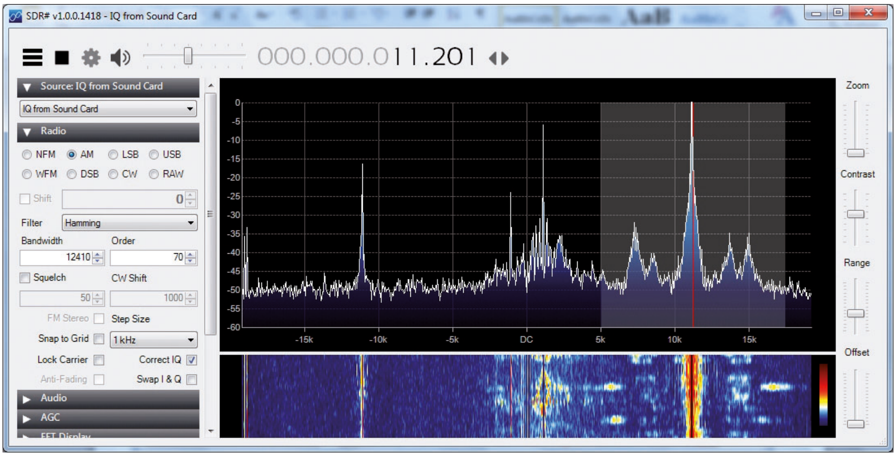

Figure 1.22: AM reception

The first test is best done with an AM broadcast transmitter. In the evening, you usually have a large selection of shortwave radio bands at your disposal. During the day, strong signals can be found especially on the higher bands. The AM mode must be selected in the radio menu. With the mouse, the AM reception range in the spectrum can be shifted to the exact frequency.

Software Defined Radio

SDR Hands-on Book

Burkhard Kainka,
Born in 1953, a Radio amateur with the callsign DK7JD. He has spent many years active as a physics teacher, and from 1996 became a self-employed developer and author of electronics and microcontroller books. He operates the websites www.elektroniklabor.de and www.b-kainka.de among other projects.

The short-wave technique has a very particular appeal: It can easily bridge long distances. By reflecting short-wave signals off the conductive layers of the ionosphere, they can be received in places beyond the horizon and therefore can reach anywhere on earth. Although technology is striving for ever higher frequencies, and radio is usually listened to on FM, DAB+, satellite or the Internet, modern means of transmission require extensive infrastructure and are extremely vulnerable. In the event of a global power outage, there is nothing more important than the short-wave. Amateur radio is not only a hobby, it's also an emergency radio system!

Elektor's SDR-Shield (article no. 18515) is a versatile shortwave receiver up to 30MHz. Using an Arduino and the appropriate software, radio stations, morse signals, SSB stations, and digital signals can be received.

In this book, successful author and enthusiastic radio amateur, Burkhard Kainka describes the modern practice of software defined radio using the Elektor SDR Shield. He not only imparts a theoretical background but also explains numerous open source software tools.

ISBN 978-1-907920-76-9

9 781907 920769

Elektor International Media BV

www.elektor.com

elektor

LEARN DESIGN SHARE