

# threeme3/QCX-SSB

Originalartikel

Backup

<html> <p>This is a simple and experimental modification that transforms a <a href=„<https://qrp-labs.com/qcx.html>“ rel=„nofollow“>QCX</a> into a (Class-E driven) SSB transceiver. It can be used to make QRP SSB contacts, or (in combination with a PC) used for the digital modes such as FT8, JS8, FT4. It can be fully-continuous tuned through bands 80m-10m in the LSB/USB-modes with a 2400Hz bandwidth has up to 5W PEP SSB output and features a software-based full Break-In VOX for fast RX/TX switching in voice and digital operations.</p><p>The SSB transmit-stage is implemented entirely in digital and software-based manner: at the heart the ATMEGA328P is sampling the input-audio and reconstructing a SSB-signal by controlling the SI5351 PLL phase (through tiny frequency changes over 800kbit/s I2C) and controlling the PA Power (through PWM on the key-shaping circuit). In this way a highly power-efficient class-E driven SSB-signal can be realized; a PWM driven class-E design keeps the SSB transceiver simple, tiny, cool, power-efficient and low-cost (ie. no need for power-inefficient and complex linear amplifier with bulky heat-sink as often is seen in SSB transceivers).</p><p>For the receiver, a large portion of the original QCX circuit has been removed and implemented in digital manner (software): the ATMEGA328P is now implementing the 90 degree phase shift circuit, the (CW/SSB) filter circuit and the audio amplifier circuit (now a class-D amplifier). This has simplified the QCX circuit a lot (50% less components needed), and there are a number of advantages and features: there is no longer a need for an alignment procedure due to the very accurate 90 degree Hilbert phase shifter; and there are now adjustable IF DSP filters for CW and SSB; and there is an AGC and there is a noise-reducing DSP signal conditioning function and there are three indepent built-in attenuators in the analog front-end which helps utilizing the full dynamic range. The speaker is directly connected and driven by the ATMEGA. A digital mixer with narrow low-pass window (2 kHz), steep roll-off (-45dB/decade) combined with an oversampling and decimating ADC are responsible for a processing gain, dynamic range and alias rejection sufficient to handle weak and strong signal conditions (e.g. contests or listening on 40m just next to broadcasting band).</p><p>This experiment is created to try out what is can be achieved with minimal hardware while moving complexity towards software; here the approach followed is to simplify the design where possible while keep a reasonable performance. The result is a cheap, easy to build, versatile QRP SSB transceiver that actually is quite suitable for making QSOs (even in contest situations), however due to the experimental nature some parts are still in progress and hence limited. Feel free to try it out or to experiment with this sketch, let me know your thoughts or contribute here: <a href=„<https://github.com/threeme3/QCX-SSB>“><https://github.com/threeme3/QCX-SSB></a> The original forum discussion on the topic here: <a href=„<https://groups.io/g/QRPLabs/topic/29572792>“ rel=„nofollow“>QRPLabs Forum</a></p><p><strong>Update: A new group with collaborative interest has started to build a compact, portable (SOTA) rig for the summer, based on the ideas presented here. More information can be found in this: <a href=„<https://groups.io/g/ucx>“ rel=„nofollow“>Forum</a>.</strong></p><p>73, Guido <a href=„mailto:pe1nnz@amsat.org“>pe1nnz@amsat.org</a></p><p><a target=„\_blank“ rel=„noopener noreferrer“ href=„<https://github.com/threeme3/QCX-SSB/blob/master/top.png>“><img src=„<https://github.com/threeme3/QCX-SSB/raw/master/top.png>“ alt=„ class=„c1“ referrerpolicy=„no-referrer“ /></a></p><h2><a id=„user-content-list-of-features“ class=„anchor“ aria-hidden=„true“ href=„#list-of-features“></a>List of features:</h2><ul><li><strong><a href=„<https://github.com/threeme3/QCX-SSB/blob/master/ucx.png>“>Simple, fun and versatile</a> QRP SSB HF transceiver</strong> with embedded <strong>DSP and SDR functions</strong>;</li>

<li><strong><a href=„<https://core.ac.uk/download/pdf/148657773.pdf>“ rel=„nofollow“>EER Class-E</a></strong> driven SSB transmit-stage</li> <li>Approximately <strong>5W PEP SSB output</strong> from 13.8V supply</li> <li><strong>All-Mode support: USB, LSB, experimental modes: CW, AM, FM</strong></li> <li><strong>DSP filters: 4000, 2500, 1700, 500, 200, 100, 50 Hz passband</strong></li> <li><strong>DSP features: Automatic Gain Control (AGC), Noise-reduction (NR), Voice-triggered Xmit (VOX), RX Attenuators (ATT), TX drive control, Volume control, dBm/S-meter.</strong></li> <li>SSB opposite side-band/carrier suppression <strong>Transmit: better than -45dBc, IMD3 (two-tone) -33dBc, Receive: better than -50dBc</strong></li>
 <li><strong>Multiband</strong> support, continuously tunable through bands <strong>80m-10m</strong> (and from 20kHz..99MHz with loss in performance)</li>
 <li><strong>Open source</strong> firmware, built with Arduino IDE; allows experimentation, new features can be added, contributions can be shared via Github, software-complexity: 2000 lines of code</li> <li>Software-based <strong>VOX</strong> that can be used as <strong>fast Full Break-In</strong> (QSK operation) or assist in RX/TX switching for operating digital modes (no CAT or PTT interface required)</li> <li><strong>Simple to install modification</strong> with <strong>8 component changes and 8 wires</strong></li> <li><strong>Lightweight and low-cost transceiver design</strong>: because of the EER-transmitter class-E stage it is <strong>highly power-efficient</strong> (no bulky heatsinks required), and has a <strong>simple design</strong> (no complex balanced linear power amplifier required)</li> <li><strong>Fully digital and software-based SSB transmit-stage</strong>: samples microphone-input and reconstruct a SSB-signal by controlling the phase of the SI5351 PLL (through tiny frequency changes over 800kbits/s I2C) and the amplitude of the PA (through PWM of the PA key-shaping circuit)</li> <li><strong>Fully digital and software-based SDR receiver-stages (optionally)</strong>: samples I/Q (complex) signal from Quadrature Sampling Detector digital mixer, and performs a 90-degree phase-shift mathematically in software (Hilbert-transform) and cancels out one side-band by adding them</li> <li>Three independent switchable analog front-end <strong>receiver attenuators (0dB, -13dB, -20dB, -33dB, -53dB, -60dB, -73dB)</strong></li> <li>Receiver Noise floor <strong>MDS: 135 dBm</strong> at 28MHz (in 200Hz BW)</li> <li>Receiver Front-end selectivity: <strong>steep -45dB/decade roll-off +/-2kHz from tuned-frequency</strong></li> <li>Blocking <strong>dynamic range: 20kHz offset 123dB, 2kHz offset 78dB</strong></li> <li><strong>CW decoder (experimental).</strong></li>
 <li>Probably the most <strong>cost effective</strong> and <strong>easy</strong> to build standalone SDR/SSB transceiver that you can find. Very much <strong>simplifies</strong> the original QCX circuit (i.e. <strong>50% less components to install</strong>, <strong>no complex transformer windings</strong>, <strong>no alignment procedure</strong>) and more <strong>versatile</strong> in use.</li>

</ul><h2><a id=„user-content-revision-history“ class=„anchor“ aria-hidden=„true“ href=„#revision-history“>Revision History:</a></h2><table><thead><tr><th>Rev.</th> <th>Date</th> <th>Features</th></tr></thead><tbody><tr><td>R1.02 (<strong>current</strong>)</td> <td>2020-04-12</td> <td>Integrated SDR receiver, CW decoder, DSP filters, AGC, NR, ATT, experimental modes CW, AM, FM, quick menu, persistent settings, improved SSB TX quality. LCD fix, selectable CW pitch.</td></tr><tr><td>R1.01d</td> <td>2019-05-05</td> <td>Q6 now digitally switched (remove C31) - improving stability and IMD. Improved signal processing, audio quality, increased bandwidth, cosmetic changes and reduced RF feedback, reduced s-meter RFI, S-meter readings, self-test on startup. Receiver I/Q calibration, (experimental) amplitude pre-distortion and calibration. <strong>See here <a href=„<https://github.com/threeme3/QCX-SSB/tree/R1.01d>“>original QCX-SSB modification</a> (it is also supported by current firmware!)</strong></td></tr><tr><td>R1.00</td> <td>2019-01-29</td> <td>Initial release of SSB transceiver prototype.</td></tr></tbody></table><h2><a id=„user-content-schematic“ class=„anchor“ aria-hidden=„true“ href=„#schematic“>Schematic:</a></h2><p>Below the schematic after the modification is applied, unused components are left out and changed components are marked in red (click to zoom & download) (link to <a href=„<https://qrp-labs.com/images/qcx/HiRes.png>“</a>)

rel=„nofollow“>original schematic</a>): <a target=„\_blank“ rel=„noopener noreferrer“ href=„<https://github.com/threeme3/QCX-SSB/blob/master/schematic.png>“><img src=„<https://github.com/threeme3/QCX-SSB/raw/master/schematic.png>“ alt=„schematic“ class=„c1“ referrerpolicy=„no-referrer“ /></a></p><h2><a id=„user-content-installation“ class=„anchor“ aria-hidden=„true“ href=„#installation“></a>Installation:</h2><p>Please find here the latest <strong><a href=„[https://github.com/threeme3/QCX-SSB/blob/master/QCX-SSB\\_assembly\\_Rev-5.pdf](https://github.com/threeme3/QCX-SSB/blob/master/QCX-SSB_assembly_Rev-5.pdf)“>PCB Rev5 Assembly Manual CLICK HERE</a> by Manuel, DL2MAN (document revision 1.1), 23-May-2020</strong>. Instruction for older PCB revisions can be found below:</p><p>This modification consists of a few component changes and wires:</p><ul><li><strong>To simplify things, 79 components are no longer needed</strong> (and you could omit them on an unbuilt QCX): IC6-10,R11-40,43,55,R59-64,C1,5,8,C9-28,C31,C52-54,L1-3,D5,Q7,T1,JP1/DVM/FREQ.</li><li><strong>SDR receiver</strong>: change R7,10(82k); C4,C7(1nF), remove R11,12,14,15,17,27,29,59,IC10; wire IC2(pin15) to IC10(pin1), insert 470R if you intend to use a headphones; C39(R27/R29 side) to IC5(pin1); C40(IC10 side) to IC5(pin7), and disconnect R50(5V side) and R52(5V side) and wire both to IC2(pin21). If you cannot wire C39,R50 then make the following Rev5 mods in addition: remove IC7, change R11(.1uF), C9(10k), R53(1k); wire 10k from R52(IC2-pin21 side) to IC7(pin5); wire R11 (C9 side) to R27(middle). If you want to build-up the original (analog) QCX receiver, you can skip this step and insert a SPDT switch between C21(+ side) and IC9(pin1) for CW and R27(pin2) for SSB (as was done in <a href=„<https://github.com/threeme3/QCX-SSB/tree/R1.01d>“>original QCX-SSB modification</a>).</li><li><strong>SSB transmitter</strong>: change D4,R56 (10k); R58 (.22uF); C32 (10uF), remove C31; wire IC2(pin21) to pin R57(side DVM-pin3); wire IC2(pin20) to DVM(pin2); wire IC2(pin18) to junction D4-C42-R58; install a microphone to tip (+ of electret) and sleeve GND of paddle-jack; PTT-switch to ring and sleeve (<a href=„[https://vignette.wikia.nocookie.net/x1m/images/f/f1/X1M\\_mic\\_pinout\\_diagram.jpg/revision/latest?cb=20131028014710](https://vignette.wikia.nocookie.net/x1m/images/f/f1/X1M_mic_pinout_diagram.jpg/revision/latest?cb=20131028014710)“ rel=„nofollow“>X1M-mic</a>).</li> <li><strong>Multiband support</strong>: remove C1,5,8,R64,T1; at T1 landing pattern (see <a href=„[https://www.qrp-labs.com/images/qcx/assembly\\_A4\\_Rev\\_4b.pdf](https://www.qrp-labs.com/images/qcx/assembly_A4_Rev_4b.pdf)“ rel=„nofollow“>QCX Assembly instruction</a> T1) install R (1k) over 6-8; R (1k) over 3-4; C (10nF) over 4-8; change C30 (30pF); L4 (1uH/16t); replace C25-28,L1-L3 with LPFs for different bands.</li></ul><p><strong>Firmware</strong>: download <a href=„<https://github.com/threeme3/QCX-SSB/releases>“>latest released hex file</a> and install with <a href=„<https://www.qrp-labs.com/qcx/qcxfirmware.html>“ rel=„nofollow“>standard QCX firmware upload procedure</a> (see also <a href=„#note1“>note 1</a>).</p><p>Below the (Rev4) layout with components marked in red that needs to be changed; gray components must be installed and blank components may be omitted and some must be remove (see above): <a target=„\_blank“ rel=„noopener noreferrer“ href=„<https://github.com/threeme3/QCX-SSB/blob/master/layout.png>“><img src=„<https://github.com/threeme3/QCX-SSB/raw/master/layout.png>“ alt=„layout“ class=„c1“ referrerpolicy=„no-referrer“ /></a></p><p>Below the (Rev4) wires that needs to be installed on the bottom PCB. Note that a circle represents resistor R50/R52 that is in series with the wire and the pad at which the circle is drawn (the resistor is no longer connected to its original neighbour pad and its probably the best to place this wiring and resistor at the components side): <a target=„\_blank“ rel=„noopener noreferrer“ href=„<https://github.com/threeme3/QCX-SSB/blob/master/pcb.png>“><img src=„<https://github.com/threeme3/QCX-SSB/raw/master/pcb.png>“ alt=„pcb“ class=„c1“ referrerpolicy=„no-referrer“ /></a></p><p>See here the completed transceiver with the (Rev4) modifications applied (includes PA bias mod <a href=„#note3“>note 3</a>, does not include LPFs): <a target=„\_blank“ rel=„noopener noreferrer“ href=„<https://github.com/threeme3/QCX-SSB/blob/master/pcb-top.png>“><img src=„<https://github.com/threeme3/QCX-SSB/raw/master/pcb-top.png>“ alt=„“ class=„c1“

referrerpolicy=„no-referrer“ /></a> <a target=„\_blank“ rel=„noopener noreferrer“ href=„<https://github.com/threeme3/QCX-SSB/blob/master/pcb-bottom.png>“><img src=„<https://github.com/threeme3/QCX-SSB/raw/master/pcb-bottom.png>“ alt=„ class=„c1“ referrerpolicy=„no-referrer“ /></a></p><p><strong>Build options:</strong></p><ol><li>For an unbuilt QCX, the easiest way to build this digital transceiver is to omit the 79 components and apply all above modifications, this will offer you all features described above. If you want to, you can still decide to build the original QCX from here.</li> <li>If you already have a working QCX, the easiest way is to apply the SSB transmit mod and bypass the CW filter with a switch, in this way your transceiver keeps the analog receiver circuit, there are no SDR features, and you can still place the original QRP Labs CW firmware if you prefer.</li> </ol><h2><a id=„user-content-operation“ class=„anchor“ aria-hidden=„true“ href=„#operation“></a>Operation:</h2><p>Currently, the following functions have been assigned to shortcut buttons (L=left, E=encoder, R=right) and menu-items:</p><table><thead><tr><th>Menu Item</th> <th>Function</th> <th>Button</th></tr></thead><tbody><tr><td>1.1 Volume</td> <td>Audio level (0..16) & power-off/on</td> <td><strong>E +turn</strong></td></tr><tr><td>1.2 Mode</td> <td>Modulation (LSB, USB, CW, AM, FM)</td> <td><strong>R</strong></td></tr><tr><td>1.3 Filter BW</td> <td>Audio passband (Full, 300..4000, 300..2500, 300..1700, 200, 100 Hz)</td> <td><strong>R double</strong></td></tr><tr><td>1.4 Band</td> <td>Band-switch to pre-defined FT8 freqs (80,60,40,30,20,17,15,12,10,6,4m)</td> <td><strong>E double</strong></td></tr><tr><td>1.5 Tuning Rate</td> <td>Tuning step size 10M, 1M, 0.5M, 100k, 10k, 1k, 0.5k, 100, 10, 1</td> <td><strong>E or E long</strong></td></tr><tr><td>1.6 AGC</td> <td>Automatic Gain Control (ON, OFF)</td> <td><strong>1.7 NR</strong></td><td>Noise-reduction level (0-8), load-pass & smooth</td></tr><tr><td>1.8 ATT</td> <td>Analog Attenuator (0, -13, -20, -33, -40, -53, -60, -73 dB)</td></tr><tr><td>1.9 ATT2</td> <td>Digital Attenuator in CIC-stage (0-16) in steps of 6dB</td></tr><tr><td>1.10 S-meter</td> <td>Type of S-Meter (OFF, dBm, S, S-bar)</td></tr><tr><td>2.1 CW Decoder</td> <td>Enable/disable CW Decoder (ON, OFF)</td></tr><tr><td>2.2 CW Tone</td> <td>CW Filter+Side-tone (300, 700)</td></tr><tr><td>2.3 CW Offset</td> <td>CW RX Offset (use to align with CW Filter Tone)</td></tr><tr><td>3.1 VOX</td> <td>Voice Operated Xmit (ON, OFF)</td><td><strong>R long</strong></td></tr><tr><td>3.2 VOX Level</td> <td>Audio threshold of VOX (0-255)</td></tr><tr><td>3.3 MOX</td> <td>Monitor on Xmit (audio unmuted during transmit)</td></tr><tr><td>3.4 TX Drive</td> <td>Transmit audio gain (0-8) in steps of 6dB, 8=constant amplitude for SSB</td></tr><tr><td>8.1 Ref freq</td> <td>Actual si5351 crystal frequency, used for frequency-calibration</td></tr><tr><td>8.2 PA Bias min</td> <td>KEY\_OUT PWM level (0-255) for representing 0% RF output</td></tr><tr><td>8.3 PA Bias max</td> <td>KEY\_OUT PWM level (0-255) for representing 100% RF output</td></tr><tr><td>9.1 Sample rate</td> <td>for debugging, testing and experimental purpose</td></tr><tr><td>9.2 CPU load</td> <td>for debugging, testing and experimental purpose</td></tr><tr><td>9.3 Param A</td> <td>for debugging, testing and experimental purpose</td></tr><tr><td>9.4 Param B</td> <td>for debugging, testing and experimental purpose</td></tr><tr><td>9.5 Param C</td> <td>for debugging, testing and experimental purpose</td></tr><tr><td>main</td> <td>Frequency (20kHz..99MHz)</td></tr><tr><td><strong>turn</strong></td> <td><tr><td>main</td> <td>Quick menu</td></tr><tr><td><strong>L +turn</strong></td> <td><tr><td>main</td> <td>Menu enter</td></tr><tr><td><strong>L</strong></td> <td>Menu back</td></tr><tr><td><strong>R</strong></td> <td></tr></tbody></table><p>Operating Instructions:</p><p>Tuning can be done by turning the rotary encoder. Its step size can be decreased or increased by a short or long press. A change of band can be done with a double press. The mode of operation is altered with a short press on the right button; a double press on right button narrows the receiver filter bandwidth, the bandwidth is reset every time mode is changed. The volume is changed by turning the rotary encoder while pressed.</p><p>There is a menu available that can be accessed by a short left press. With the encoder it is possible to navigate through this menu. When you want to change a menu

parameter, a press with left button allows you to change the parameter with the encoder. With the right button it is possible to exit the menu any time. A fast access to the menu and parameter can be achieved by pressing the left button while turning the encoder, once you lift the left button you can immediately change the parameter by turning the encoder. For receive, by default an AGC is enabled. This increases the volume when there are weak signals and decreases for strong signals. This is good for SSB signals but can be annoying for CW operation. The AGC can be turned off in the menu, this makes the receiver less noisy but require more manual volume change. To further reduce the noise, a noise-reduction function can be enabled in the menu with the NR parameter. To use the available dynamic range optimally, you can attenuate incoming signal by enabling a front-end attenuator with „ATT“ parameter. Especially on frequencies 3.5-7 MHz the atmospheric noise levels are much higher, so you can increase the receiver performance by adding attenuation (e.g 13dB) such that the noise-floor is still audible. To calibrate the transceiver frequency, you can tune to a calibrated signal source (e.g. WWV on 10 MHz) and zero-beat the signal by changing „Ref freq“ parameter; alternatively you can measure the XTal frequency with a counter and set the parameter. A S-meter of choice (dBm, S, S-bar) can be selected with the S-meter parameter. Selecting an S-bar, shows a signal-strength bar where each tick represents a S-point (6dB). For SSB voice operation, connect a microphone to the paddle jack, a PTT or onboard „key“ press will bring the transceiver into transmit. With the „TX Drive“ parameter, it is possible to set the modulation depth or PA drive, it is default set to 4 increasing it gives a bit more punch (compression for SSB). Setting it to a value 8 in SSB means that the SSB modulation is transmitted with a constant amplitude (possibly reducing RFI but at the cost of audio quality). To monitor your own modulation, you can temporarily increase MOX parameter. A long press on the right button enters the transceiver in VOX operation, the VOX sensitivity can be configured in the menu with „VOX threshold“ parameter. The PA Bias min and max parameters sets the working range of the PWM envelope signal, a range of 0-255 is the full range which is fine if you use a key-shaping circuit for envelope control, but when you directly bias the PA MOSFETs ([note 3](#note3)) with the PWM signal then you specify the optimal working range from just above the MOSFET threshold level to the maximum peak power you would like to use (0-180 are good values on my QCX). For FT8 (and any other digital) operation, select one of the pre-programmed FT8 bands by double press the rotary encoder, connect the headphone jack to sound card microphone jack, sound card speaker jack to microphone jack, and give a long press on right button to enter VOX mode. Adjust the volume to a minimum and start your favorite FT8 application (JTDX for instance). The sensitivity of the VOX can be set in the „VOX threshold“ parameter. On startup, the transceiver is performing a self-test. It is checking the supply and bias voltages, I2C communications and algorithmic performance. In case of deviations, the display will report an error during startup. It also discovers the capabilities of the transceiver depending on the mods made. The following capabilities are detected and shown on the display: „QCX“ for a QCX without mods; „QCX-SSB“ for a QCX with SSB mod; „QCX-DSP“ for a QCX with SIDETONE disconnected and connected to a speaker (through decoupling capacitor); „QCX-SDR“ for a QCX with SDR mod. Please check if this capability matches with the mods.

[Technical Description:](#technical-description)

The principle of operation (at least what is going on in the ATMEGA) is a bit as in the following video-fragment: [Opzij](https://youtu.be/uN706PiFLm0) (in Dutch saying; „Sideways, sideways, sideways, Make room, make room, make room, We are in an incredible rush“); full [lyrics](https://www.google.com/search?q=lyrics+opzij)



here... jokes aside; below the block diagram of the QCX-SSB, SDR transceiver: [here](https://github.com/threeme3/QCX-SSB/blob/master/block.png)

[here](https://github.com/threeme3/QCX-SSB/raw/master/block.png)

For SSB reception, the QCX analog phasing receiver stage is replaced with a digital SDR stage; this means that the phase shifting op-amp IC6 is changed

into a regular amplifier and whereby the individual I and Q outputs are directly fed into the ATMEGA328P ADC inputs for signal processing. The ATMEGA328P (over-)samples the ADC input at a 62kHz sample-rate, an decimates this high-samplerate to a lower samplerate, performs a phase-shift by means of a Hilbert-transform, summing the result to obtain side-band rejection; it subsequently applies a low-pass filtering, AGC and noise-reduction functions. Since the original QCX phase-shifting network and analog CW filter are not used, about half of the original QCX components can be left out; by combining the function of IC7B into IC6A another op-amp can be saved. The ADC inputs are low-pass filtered (-40dB/decade roll-off at 1.5kHz cut-off) to prevent aliasing and input are biased with a 1.1V analog reference voltage to obtain additional sensitivity and dynamic range. With the 10-bit ADCs and a 4x over-sampling rate, a theoretical dynamic range of 72dB can be obtained in 2.4kHz SSB bandwidth. LSB/USB mode switching is done by changing the 90 degree phase shift on the CLK0/CLK1 signals of the SI5351 PLL. Three embedded attenuators are available for optimally using dynamic range; the first attenuator is the RX MOSFET switch Q6 responsible for 20dB attenuation, the second attenuator is ADC range (1.1V or 5V) selected by the ATMEGA ADC analog reference (AREF) logic and is responsible for 13dB attenuation, the third attenuator is a pull-down of an analog input on the ATMEGA with a GPIO port responsible for 53dB attenuation. Combining the three attenuators provides the attenuation steps 0dB, -13dB, -20dB, -33dB, -53dB, -60dB, -73dB.

For SSB transmission the QCX DVM-circuitry is changed and used as an audio-input circuit. An electret-microphone (with PTT switch) is added to the Paddle jack connecting the DVM-circuitry, whereby the DOT input acts as the PTT and the DASH input acts as the audio-input. The electret microphone is biased with 5V through a 10K resistor. A 10nF blocking capacitor prevents RF leakage into the circuit. The audio is fed into ADC2 input of the ATMEGA328P microprocessor through a 220nF decoupling capacitor. The ADC2 input is biased at 0.55V via a divider network of 10K to a 1.1V analog reference voltage, with 10-bits ADC resolution this means the microphone-input sensitivity is about 1mV (1.1V/1024) which is just sufficient to process unamplified speech.

A new QCX-SSB firmware is uploaded to the ATMEGA328P, and facilitates a [digital SSB generation technique](http://pe1nnz.nl.eu.org/2013/05/direct-ssb-generation-on-pll.html) in a completely software-based manner. A DSP algorithm samples the ADC2 audio-input at a rate of 4x4800 samples/s, performs a Hilbert transformation and determines the phase and amplitude of the complex-signal; the phase-changes are restricted<sup>[note 2](#note2)</sup> and transformed into either positive (for USB) or negative (for LSB) phase changes which in turn transformed into temporary frequency changes which are sent 4800 times per second over 800kbit/s I2C towards the SI5351 PLL. This result in phase changes on the SSB carrier signal and delivers a SSB-signal with a bandwidth of 2400 Hz whereby spurious in the opposite side-band components is attenuated.

The amplitude of the complex-signal controls the supply-voltage of the PA, and thus the envelope of the SSB-signal. The key-shaping circuit is controlled with a 32kHz PWM signal, which can control the PA voltage from 0 to about 12V in 256 steps, providing a dynamic range of



$(\log_2(256) * 6)$  48dB in the SSB signal. C31 is removed to ensure that Q6 is operating as a digital switch, this improves the efficiency, thermal stability, linearity, dynamic range and response-time.

Though the amplitude information is not mandatory to make a SSB signal intelligible, adding amplitude information improves quality. The complex-amplitude is also used in VOX-mode to determine when RX and TX transitions are supposed to be made. Instead of using a key-shaping circuit for envelope control, it is possible to directly bias the PA MOSFETs with the (filtered) PWM signal. This has the advantage of less losses and simplifies at the cost of linearity which result in more compression for an SSB signal (which is actually a good thing).

The IMD performance is related dependent on the quality of the system: the linearity (accuracy) of the amplitude and phase response and the precision (dynamic range) of these quantities. Especially the DSP bit-width, the precision used in the DSP algorithms, the PWM and key-shaping circuit that supplies the PA and the PA phase response are critical. Decreasing (or removing) C32 improves the IMD characteristics but at the cost of an increase of PWM products around the carrier.

class=„anchor“ aria-hidden=„true“ href=„#results“></a>Results</h2><p>Here is a <a href=„<https://youtu.be/-QfMQuIk0eA>“ rel=„nofollow“>sample</a> me calling CQ on 40m with my QCX-SSB at 5W and received back by the Hack Green websdr about 400km away, note that the audio quality has further improved since then.</p><p>Several OMs reported a successful QCX-SSB modification and were able to make SSB QRP DX contacts over thousands of kilometers on the 20m and 40m bands. During CQ WW contest I was able to make 34 random QSOs on 40m with 5W and an inverted-V over the house in just a few hours with CN3A as my furthest contact, I could observe the benefits of using SSB with constant-envelope in cases where my signal was weak; for FT8 a Raspberry Pi 3B+ with JTDX was used to make FT8 contacts all the way up to NA.</p><p>Measurements: The following performance measurements were made with QCX-SSB R1.01, a modified RTL-SDR, Spektrum-SVmod-v0.19, Sweex 5.0 USB Audio device and Audicity player. It is recognized that this measurement setup has its own limitations, hence the dynamic range of the measurements is somewhat limited by the RTL-SDR as this device goes easily into overload. Measurements were made with the following settings: USB modulation, no pre-distortion, two-tone input 1000Hz/1200Hz where audio level is set just before the point where compression starts.

Results:</p><ul><li>Intermodulation distortion products (two-tone; SSB with varying envelope) IMD3, IMD5, IMD7: respectively -33dBc; -36dBc; -39dBc</li> <li>Intermodulation distortion products (two-tone; SSB with constant envelope) IMD3, IMD5, IMD7: respectively -16dBc; -16dBc; -19dBc</li> <li>Opposite side-band rejection (two-tone): better than -45dBc</li> <li>Carrier rejection (two-tone): better than -45dBc</li> <li>Wide-band spurious (two-tone): better than -45dBc</li> <li>3dB bandwidth (sweep): 0..2400Hz <a target=„\_blank“ rel=„noopener noreferrer“ href=„<https://github.com/threeme3/QCX-SSB/blob/master/twotone.png>“><img src=„<https://github.com/threeme3/QCX-SSB/raw/master/twotone.png>“ alt=„twotone“ class=„c1“ referrerpolicy=„no-referrer“ /></a></li> </ul><h3><a id=„user-content-notes“ class=„anchor“ aria-hidden=„true“ href=„#notes“></a>Notes:</h3><ol><li><a name=„user-content-note1“>Firmware upload variations:</a></li> </ol><ul><li><a href=„<https://qrp-labs.com/images/qcx/HowToUpdateTheFirmwareOnTheQCXusingAnArduinoUNOandAVRDUDUDESS.pdf>“ rel=„nofollow“>Arduino as ISP</a> instructions if you have a Arduino UNO board available (tip: use female-to-male breadboard cables to connect Arduino to QCX ISP jumper); or <a href=„[https://sites.google.com/site/g4zfqradio/qrplabs\\_program\\_chip\\_with\\_USBasp](https://sites.google.com/site/g4zfqradio/qrplabs_program_chip_with_USBasp)“ rel=„nofollow“>USBasp</a> instructions if you have a USBasp programmer, alternatively use <a href=„<https://groups.io/g/QRPLabs/topic/57461404#40024>“ rel=„nofollow“>USPasp ExtremeBurner</a>; but many other ISP programmers can be used in similar manner such as <a href=„<https://groups.io/g/QRPLabs/attachment/40315/0/QCX%20Firmware%20Update%20Instructions.pdf>“ rel=„nofollow“>USBtiny</a> or AVRisp mkII. During ISP, mic should be disconnected, power supply should be connected; in tool do not erase, program EEPROM or set fuse settings (they are by default ok: E=FD H=D6 L=FF).</li> <li>Alternatively, in case you have an ATMEGA328P chip with Arduino bootloader, you can place the chip in an Arduino UNO board and upload directly (without the need for a ISP cable and QCX) by specifying 'arduino' programmer and baudrate 115200.</li> <li>Alternatively, in case you have an <a href=„<https://www.arduino.cc/en/Main/OldSoftwareReleases#previous>“ rel=„nofollow“>Arduino 1.8.9</a> (or newer) environment installed, you can upload the <a href=„<https://github.com/threeme3/QCX-SSB/blob/master/QCX-SSB.ino>“>QCX-SSB Sketch</a> directly from the Arduino environment (without using AVRDUDESS and firmware file); make sure „Tools &gt; Board &gt; Arduino/Genuino Uno“, „Tools &gt; Port &gt; /dev/ttyUSB0 or ttyACM0“, and then „Sketch &gt; Upload“ is selected, while the ATMEGA328P chip is placed in the Arduino UNO socket. It is also possible to use <a

href=„<https://qrp-labs.com/images/qcx/HowToUpdateTheFirmwareOnTheQCXusingAnArduinoUNOandAVRDUDESS.pdf>“ rel=„nofollow“>Arduino as ISP</a> method: upload this variation of <a href=„<https://raw.githubusercontent.com/adafruit/ArduinoISP/master/ArduinoISP.ino>“ rel=„nofollow“>ArduinoISP</a> to the Arduino board and select „Tools &gt; Programmer &gt; Arduino as ISP“, and „Sketch &gt; Upload Using Programmer“. </li> </ul><ol start=„2“><li><a name=„user-content-note2“>The occupied SSB bandwidth can be further reduced by restricting the maximum phase change (set MAX\_DP to half a unit-circle \_UA/2 (equivalent to 180 degrees)). Audio-input can be attenuated by increasing parameter MIC\_ATTEN (6dB per step). </a></li> <li><a name=„user-content-note3“>Alternatively, the PA MOSFETs can be directly biased by the PWM envelope signal, basically making the key-shaping circuit redundant. To do so, Q6,Q4,R41,R42,C32,C31 can be removed entirely, whereby C-E pads of Q6 are wired, and where a 10nF capacitor is inserted at IC3A-pin3 and G of Q1-3, and where a 10k resistor is placed at G-D pads of Q4, a 10nF capacitor between S-D pads of Q4, and where a 10k resistor is placed between D of Q4 and G of Q1-3.</a></li> </ol><a id=„user-content-credits“ class=„anchor“ aria-hidden=„true“ href=„#credits“></a>Credits:<p><a href=„<https://qrp-labs.com/qcx.html>“ rel=„nofollow“>QCX</a> (QRP Labs CW Xcvr) is a kit designed by <em>Hans Summers (G0UPL)</em>, originally built for RSGB's YOTA summer camp 2017, a high performance, image rejecting DC transceiver; basically a simplified implementation of the <a href=„<http://www.norcalqrp.org/nc2030.htm>“ rel=„nofollow“>NorCal 2030</a> by <em>Dan Tayloe (N7VE)</em> designed in 2004 combined with a <a href=„<http://www.4sqrp.com/hipermite.php>“ rel=„nofollow“>Hi-Per-Mite</a> Active Audio CW Filter by <em>David Cripe (NM#216;S)</em>, <a href=„[http://www.gqrp.com/harmonic\\_filters.pdf](http://www.gqrp.com/harmonic_filters.pdf)“ rel=„nofollow“>Low Pass Filters</a> from <em>Ed (W3NQN)</em> 1983 Articles, a key-shaping circuit by <em>Donald Huff (W6JL)</em>, a BS170 switched <a href=„[http://www.ka7oei.com/mpm\\_class\\_e.html](http://www.ka7oei.com/mpm_class_e.html)“ rel=„nofollow“>CMOS driven MOSFET PA</a> architecture as used in the <a href=„[https://groups.yahoo.com/neo/groups/AT\\_Sprint/files/AT%20Sprint%20/](https://groups.yahoo.com/neo/groups/AT_Sprint/files/AT%20Sprint%20/)“ rel=„nofollow“>ATS</a> designs by <em>Steven Weber (KD1JV)</em> (originating from the <a href=„<https://archive.org/details/VMOSSiliconixOCR/page/n17>“ rel=„nofollow“>Power MOSFET revolution</a> in the mid 70s), and combined with popular components such as Atmel <a href=„<http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>“ rel=„nofollow“>ATMEGA328P</a> microprocessor, a Hitachi <a href=„<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>“ rel=„nofollow“>HD44780</a> LCD display and a Silicon Labs <a href=„<https://www.silabs.com/documents/public/application-notes/AN619.pdf>“ rel=„nofollow“>SI5351</a> Clock Generator (and using a <a href=„[https://www.silabs.com/community/timing/forum.topic.html/difficulty\\_settingp-LchG](https://www.silabs.com/community/timing/forum.topic.html/difficulty_settingp-LchG)“ rel=„nofollow“>phase shift in the SI5351 clocks</a>). The <a href=„<https://github.com/threeme3/QCX-SSB>“>QCX-SSB</a> transmitter and QCX-SDR receiver stage both running on a ATMEGA328P, including its multiband front-end and direct PA biasing/envelope-generation technique; its concept, circuit, code and modification to run on a QCX are a design by <em>Guido (PE1NNZ)</em>; the software-based SSB transmit stage is a derivate of earlier experiments with a <a href=„<http://pe1nnz.nl.eu.org/2013/05/direct-ssb-generation-on-pll.html>“ rel=„nofollow“>digital SSB generation technique</a> on a Raspberry Pi. The assembly manual for the QCX Rev 5 PCB, is the work of <em>Manuel (DL2MAN)</em>. Many thanks to all of you who got interested in this project and took the challege and effort to try out QCX-SSB. Without your valuable feedback and contributions the project could not have kept moving on, improving and challenging new ideas!</p> </html>

From:  
<https://schnipsl.qgelm.de/> - **Qgelm**



Permanent link:  
[https://schnipsl.qgelm.de/doku.php?id=wallabag:threeme3\\_qcx-ssb](https://schnipsl.qgelm.de/doku.php?id=wallabag:threeme3_qcx-ssb)

Last update: **2021/12/06 15:24**