Bernhard Hochstätter
Zur Steige 7
D-97522 Sand am Main

Mar 2015

# FROG-2

## Blueprint for a good QRP-CW-Sensor-Electronic Keyer with PIC16F690 microcontroller

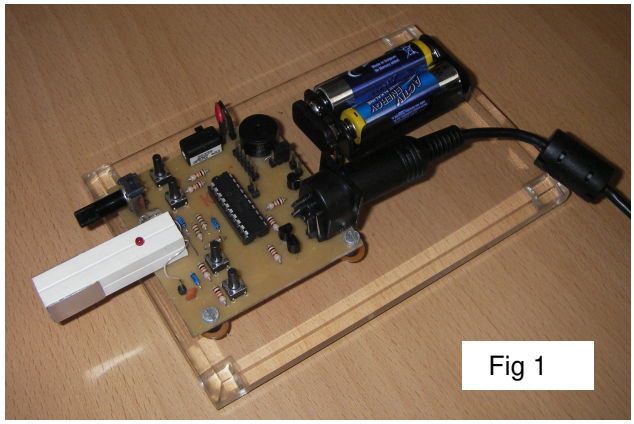


Fig 1

"Homemade" and "knowledge sharing" are important facets of Amateur-Radio for me. I was looking on the internet for a good CW-Sensor-Keyer´s source. I could not find anything useful. There are pre-programmed microcontroller offerd with / or without kit, but nothing did justify my expectations concerning price / performance.

DIY (do it yourself) was announced.
My concept approaches were low cost and functionality in daily operation.
I have programming experience in Assembler, C + + and BASIC variants. In the recent past I realized some projects with Amtel-controllers. It was tempting to do something useful with Microchip controllers, so the decision was "PIC".

Unfortunately I could not find a powerful free-compiler for PICs (MPLAB -> to expensive), so I worked for some hours with GCB or GreatCowBasic. The first result was FROG-1, a sensor-keyer with PIC12F683.
Unfortunately, some useful features were missing as the Flash (2k) was filled, so I switched to the PIC16F690. He has a larger flash (4k) and with 20 legs enough programmable inputs and outputs, FROG-2 was build up. The component costs were less than 15 €.

A components-cart exists at Reichelt-Elektronik-Company, link below:

http://www.reichelt.de/?ACTION=20;AWKID=954365;PROVID=2084

The firmware for the FROG-2 (HEX code) you get for free from me by e-mail. Contact me by e-mail, my address is dl6nbs@t-online.de

Comments and suggestions are welcome. Have fun with the construction and operation!

73 + 55 de DL6NBS
Bernd

# *CONTENT*



## *1. Features FROG-2*

- With three signal outputs, the FROG-2 is both, HSC sensor-key and QRP-electronic-keyer.
- The speed is adjustable between < 5 Wpm and HSC (High Speed Cw), by potentiometer and a digital component.
- The keyer works with 0-force sensors, no expensive mechanics necessary.
- Four memories with 61 characters each. Direct access by buttons. Memory content stays retained after power down. .
- Comfortable memory input with repetition of the last word in aberration.
- Command-set embeddable in memory text.
- Three selectable Iambic modes, no dot-dash memory, dot and dash memory, only dot memory. The mode is permanently saved.
- Adjustable Dash / Dot ratio by Dash-stretching. The selected ratio stays persistent.
- Adjustable digital-speed by menu or memory-command.
- 4-digit contest-counter adjustable by menu. Decrement by memory-command or push-button.
- XT on / off for tuning.
- Automatic power-saving (nanoamps).
- ICSP interface on board.
- Plan for a small housing.
- Firmware (hex.file) for free available at dl6nbs@darc.de

## *2. Connection and turn on the FROG-2*

The FROG-2 should be connected to the XCVR before turning on (ground potential). When switching on (slide switch left side) the FROG-2 calibrate his sensors. This process takes only milliseconds and is completed after the issue of ther version number.
Please do not touch the sensors until there!

Chapter 3, makes you now familiar with the setting of the FROG-2.

## *3. Setup FROG-2*

Simultaneous pushing button 1 and 2 enters the setup-mode. This is indicated with "S" and a short LED-flash. FROG-2 is in the setup-mode, the following settings are possible:

**Monitor - output**

Input **M** -> FROG-2 responds with:
0 = Speaker or Beeper off or 1 = Speaker or Beeper on.
Then FROG-2 signals "E" and jumps immediately back in QSO-mode.

**Contest - number**

Input **N** -> FROG-2 responds with 4-digit number and waits for a new number.
Enter the four-digit number sequence consecutive as whole word. The LED flashes briefly after each character. FROG-2 responds with "Döt" after the input sequence.
If T is used instead of 0, the output is also T instead of 0 and leading zeros of the tens and hundreds place are supplemented as T. If 0 is entered, the output is also 0 but without leading zeros. The contest counter is stored in RAM and is also available after sleep and maintained until current shutdown. Entry **E** -> Exit, return to the setup menu without changing the counter.

**Iambic - mode**

Input **C** -> FROG-2 responds with 0, 1 or 2
0 neither Dot nor Dash storage
1 with Dot memory only
2 with Dot and Dash memory
Re-enter **C** to toggle. The selected mode is saved permanently. Default after reset is 2.

**Digi - speed**

Input **S** -> FROG-2 responds continuously dit-dah
Dash-sensor speed is reduced
Dot-sensor speed is increased
Pushing button 3 will store the current value and the FROG-2 returns to setup-mode..
*Remark:*
Digi-speed sets the "working point" of the keyer. After initial start-up or reset, the potentiometer covers 7 to 25 Wpm. At Digi-speed-minimum these are 5 to 9.6 Wpm. At Digi-speed-maximum Wpm ranges from 11 up to the HSC.

**Ratio - Dash / Dot**

Input **R** -> FROG-2 responds continuously dit-dah
Dash-sensor Dash - stretching up to 6 times
Dot-sensor Dash - compression (until standard)
With button 3, the selected value is permanently stored and the FROG-2 returns to setup-mode

Pushing button 4 finishes the setup-mode. FROG-2 reports "Döt" and junps to QSO-mode. (After input "M" the backjump to QSO-mode happens automatically).

Chapter 4, makes you now familiar with the FROG-2 operation.

## *4. Operation with the FROG-2*

**Memory - input**

If button 1, 2, 3 or 4 is depressed for about 1.5 seconds, FROG-2 responds with "M". The memory is open for text entry. There is sufficient time between characters and words. The LED flashes briefly after each accepted character and signals the readiness to receive the next one. Invalid characters are detected and not accepted, the LED remains dark. Please wait at the end of a word for the "Döt" report, a word-space is addet to the last character. The capacity of each memory is 61 characters, word-spaces are not counted as consumed space. Allowed characters are A to Z 0 to 9 + - . /: = and ?
If you made a mistake, wait for the short LED-flash ang give aberration (more than 7 dots). The last word will be deleted, the program returns to the preceding word and repeats it. If aberration is keyed again, the process is repeated. Short pushing the memory-button ends the save operation with "Döt".

**Memory - recall**

Briefly pressing and releasing the buttons 1 to 4 triggers the memory readout. Repeatedly pushing a button stacks the replays. If a sensor is touched, the replay ends after the current character and all repetitions are deleted.

**Decrement - manually**

Simultaneous actuation of button 3 and 4 causes subtraction by 1 if the contest-counter is higher than zero. Only the counter is edited, there is no output to the TX. The LED lights up and the counter value is issued by the speaker.

**TX-Tuning**

Simultaneous actuation of button 2 and 3 switches the TX-on, slightly longer touching a sensor switches the TX-off.

**Embeddable commands**

Commands always start with the colon (---...)

| | |
|---|---|
| :+ | Increment counter and output the new value. |
| := | Repeat counter. |
| :Un | Increase the speed n-times (n = 1 - 9). |
| :Dn | Reduce the speed by n points (n = 1 - 9). |
| :n | Repetition of the following word n-times (n = 1 - 9). |
| :W | Wait for manual input with automatic continuation. |
| :Rn | Loop memory after n seconds (n = 1 - 9) or 3 minutes (n = 0). |
| :R | Loop memory wihout delay. |

*(Exemple 1)* :3 CQ DE :2 DL6NBS CQ PSE K :R9
CQ CQ CQ DE DL6NBS DL6NBS CQ PSE K repeat memory after 9 sec.

*(Exemple 2)* TU UR :U9 599 :+
TU UR (speed up 9-times) 599 (speed normal) current number *

** see chapter Contest Number*
*Supposition NNNN is 0007 -> If T-select: output is TT7 -> If 0-select: output is 7*
*Supposition NNNN is 2009 -> If T-select: output is 2TT9 -> If 0-select: output is 2009*

*(Exemple 3)* UR RST IS :W = NAME IS :D5 BERND
After RST IS the FROG-2 repeats "Döt" and the LED lights. Now a manual input, of any lenght, can take place. Each sensor operation sets the wait timer to zero. If no entry occurs FROG-2 continues automaticly. After the manual input = NAME IS will be sent and then, 5 points slower, BERND. The pause length is dynamic pending of the set speed.

**Sleep and wakeup**
After one hour of inactivity, the FROG-2 switches to power saving mode. If the button 1 (WAK) is shortly pressed, the FROG-2 is back with version number and is again ready for operation. The sensors are recalibrated at each wakeup.

**Reset**
Actuating button 4 and power-up resets the processor. The EPROM will be deleted, all parameters are set to the initial state. Memory contents are lost. FROG-2 emits "Döt" followed by version number. At first use a reset will be executed!

Chapter 5, makes you now familiar with technical details.

## *5. Technical details FROG-2*

**Operation as pure sensor-key**
The dot-signal is taken at T2, the dash-signal is taken at T3 (see schema). The outputs are extremely fast and can be fed to the keyer-input of the transceiver.

**Speaker or beeper**
The FROG-2 piezo-speaker is normally used for settings. The hoarse tone does not disturb, I feel it to be pleasant. The K15-jumper is set to 1-2. When replacing the speaker by a piezo beeper K15-jumper is set 2-3. The beeper will now receive the logic level and no sound signal. Beepers have high noise levels, by inserting the resistor R10 (about 150 Ohm) or a silicium-diode, the level can be slightly adjusted. When using a speaker a short piece of hose between the speaker and the sound holes amplifies the volume significantly (Fig. 8). If no resistor R10 is used, a wire-bridge instead of R10 is soldered.

**Operating voltage and current-save**
The operating voltage of the PIC12F683 may be between 2 and 5.5 V, not more! If FROG-2 is in QSO-mode (no setup) and while 60 minutes neither touches a sensor or a key is pressed, the FROG-2 goes into idle-state. Shortly before, he gives a "Döt". Depending on the operating voltage, the circuit is satisfied with some nano-amps in idle state. Actuating button 1 wakes up the FROG-2. Here are some measured power consumption values:
Ub = 2.4 V -> active: 0.82 / Sleep: 0.13
Ub = 3.0 V -> active: 0.91 / Sleep: 0.16 *
Ub = 4.8 V -> active: 2.90 / Sleep: 1.40
* Calculated operating time with 2 good AA cells: > 80 to > 500 days

**Connections**
If the 5-pin DIN connector is used: 1 = DOT, 2 = GND, 3 = DASH, 5 = Keyer.
When using the proposed housing, the partition around the area of the socket is removed. For connecting, only the 5-pin contact-tray is used (Fig. 8). The cable-tail has enough space above the battery holder. I use 20 cm shielded USB cable as terminal-tail. One end with 5-pin plug and 5-pin-socket on the tail end. From there I use adapters for each TX. When using unshielded cable problems with higher RF-power may occur, but never in QRP.

## The program

The tasks "sensor-aquisition" and "memory-handling" are running nearly parallel. The control of both tasks is realized by timer interrupts. In this way, the sound (pin 15) will be "chopped" and gets froggy, so I baptized the keyer FROG.

Sensor principle:
At PIC´s pin 5 PWM-pulses with 8 kHz are continuously generated. The pulses arrive by high impedance as a sawtooth at pin 6 and 7 and are counted during a defined period. In contact with a sensor, the signals are attenuated or completely absent and missing at the counter. The program discovers a sensor-touch.

For a reliable operation the keyer should always be connected to the radio device, thus, a sufficient mass is ensured.

Board (holes):
Parts = 0.8 mm. Potentiometer, switch, DIN-connector, beeper = 1mm. Poti-housing 2mm, mounting = 3.5mm.

DL6NBS

| **Parts** | |
|---|---|
| 2 | Resistor 0,1W / 1 M-Ohm |
| 7 | Resistor 0,1W / 10 K-Ohm |
| 1 | Resistor 0,1W / 150 Ohm |
| 4 | BAT 41 Schottky Diode 100V 0,1A |
| 3 | Transistor BC 547 |
| 1 | Housing 123x72x39mm, 4AA (optionally) |
| 1 | IC-Socket, 20-pol |
| 1 | Battery-holder for 2 x AA celles |
| 2 | Ceramic-capacitor 1,0 nF |
| 4 | Ceramic-capacitor 330 pF pF |
| 3 | Choke 47µH |
| 1 | Jumper |
| 1 | Wire-Bridge or R10 |
| 1 | LED, 3mm |
| 1 | DIN-Socket, 8-pol, print mounting |
| 1 | Chain terminals 2,54 mm, 1X24 |
| 1 | PIC16F690 DIL-20 (+Firmware DL6NBS) |
| 1 | ALPS poti. linear, 6mm, mono, 50K |
| 10 cm | 2:1 Shrink-tube, 3,2mm |
| 1 | Switch gew. RM5,08 1x ON - ON |
| 1 | Piezo-speaker EPM 121 |
| 6 | Pushbutton 6x6mm, h: 12,5mm, 12V, vert. |
| 2 | Washer 6 mm ext. 12 mm (sensors) |
| 4 | Cylinderhead screw M3 x 20 mm |
| 5 cm | Square-profile PVC 11,5 x 11,5 mm |
| 1 | Adhesive pad |

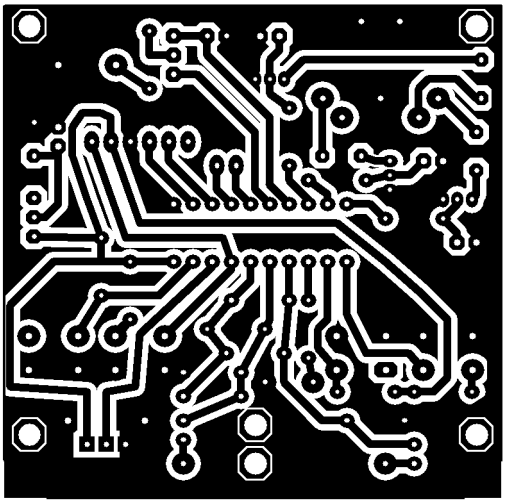


Fig 2 Circuit (Top view)

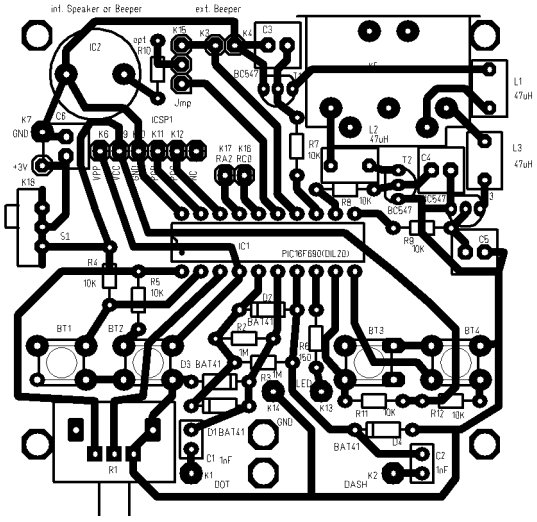
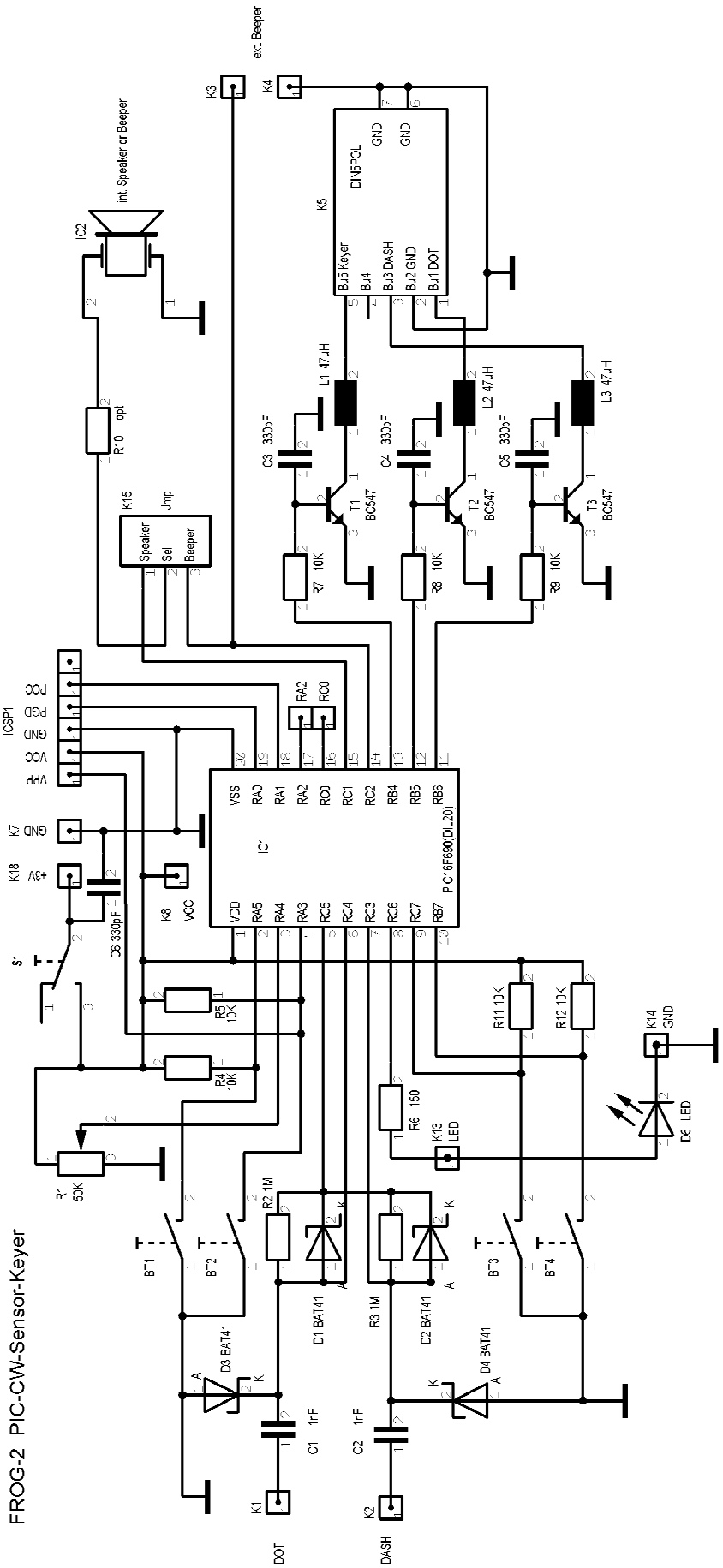



Fig 3 Parts


FROG-2 PIC-CW-Sensor-Keyer
ext. Beeper
int. Speaker or Beeper
DIN5POL
Bu5 Keyer
Bu4
Bu3 DASH
Bu2 GND
Bu1 DOT
GND
L1 47uH
L2 47uH
L3 47uH
C3 330pF
C4 330pF
C5 330pF
T1 BC547
T2 BC547
T3 BC547
R7 10K
R8 10K
R9 10K
R10 opt
Speaker
Sel
Beeper
Jmp
ICSP1
VPP
VCC
GND
PGD
PGC
PIC16F690(DIL20)
VSS
VDD
RA0
RA1
RA2
RA3
RA4
RA5
RC0
RC1
RC2
RC3
RC4
RC5
RC6
RC7
RB4
RB5
RB6
RB7
+3V
C6 330pF
S1
R5 10K
R4 10K
R11 10K
R12 10K
R6 150
LED
R1 50K
R2 1M
R3 1M
BT1
BT2
BT3
BT4
D1 BAT41
D2 BAT41
D3 BAT41
D4 BAT41
C1 1nF
C2 1nF
DOT
DASH

## *6. Mechanics and housing FROG-2 (proposal)*

The FROG-2 circuit was also designed to fit a small housing GEH KSB 02B (Fig. 5) from Reichelt Company. Figure 6 shows a dimension-plan for the processing the housing-cover. The recesses for the sensor-support, the switch and the poti-shaft is realized with saw, pincer and file after preassembly of all components by adaptation.

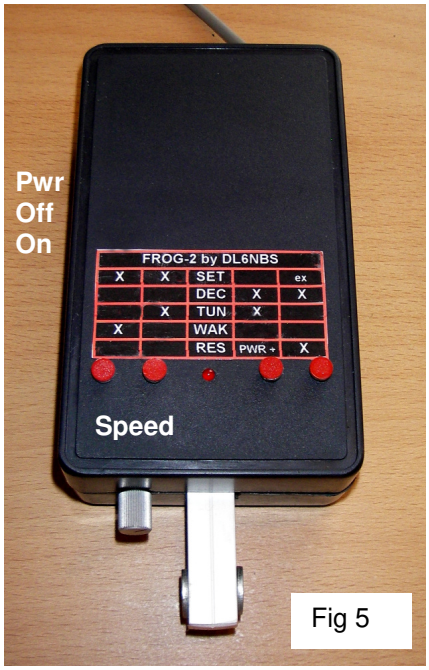


Fig 5

Two washers with 12mm diameter or better two metal plates serve as sensors. A piece of wrap-wire is soldered to the back of the washer (Figure 7a / 7b). The wires pass through 1mm holes into the interior of the 50 mm long hollow profile and are connected with the Dot- and Dash-pad on the board (Figure 8).

The 11.5 x 11.5 mm rigid PVC profile can be found at the hardware store for about 2 Euros (Alfer 21,204). The washers were associated with a drop of superglue to the profile. There are two holes on the board for fixing the sensor carrier. In my case, I have provided the two holes in the PVC profile with 3mm thread, two small metal screws to do the same. For space reasons, the PVC support is slightly shifted to the right.

Four-cylinder head screws M3 x 20 mm, in combination with four sections 3.2 mm shrink-tubing with 25 mm length, serve as button-tappet. The shrink-hoses are pushed over the thread and then carefully inserted into the cover where they slide on the tappets of the microswitches.
I treated a plastic strip with a hole puncher. If you take four of the resulting confetti chips plus a touch of superglue, ordinary head screws become handsome button-tappets (Figure 9).
A glued function-matrix simplifies operation (Figure 10).

If necessary, small sound outlet holes for the piezo speaker are drilled. In my case, a short piece of hose, as a "sound channel" between the mini-speakers and the sound holes, brought a volume-gain. If a piezo-beeper is used, no sound holes are neccessary, beepers are noisy enough.

Sensor-touching requires zero force. On smooth surfaces, non-slip-feet give sufficient stability. I use 1,5 x 1.5 cm large sections of anti-slip pad, it sticks like screwed. An additional heavy plate under the housing is also possible. This all depends finally from the operators temperament.

The mechanically savvy OM has certainly his own ideas for mechanical equipment. Figure 1 shows my FROG-2, mounted on a 10 mm plexiglass plate.

DL6NBS

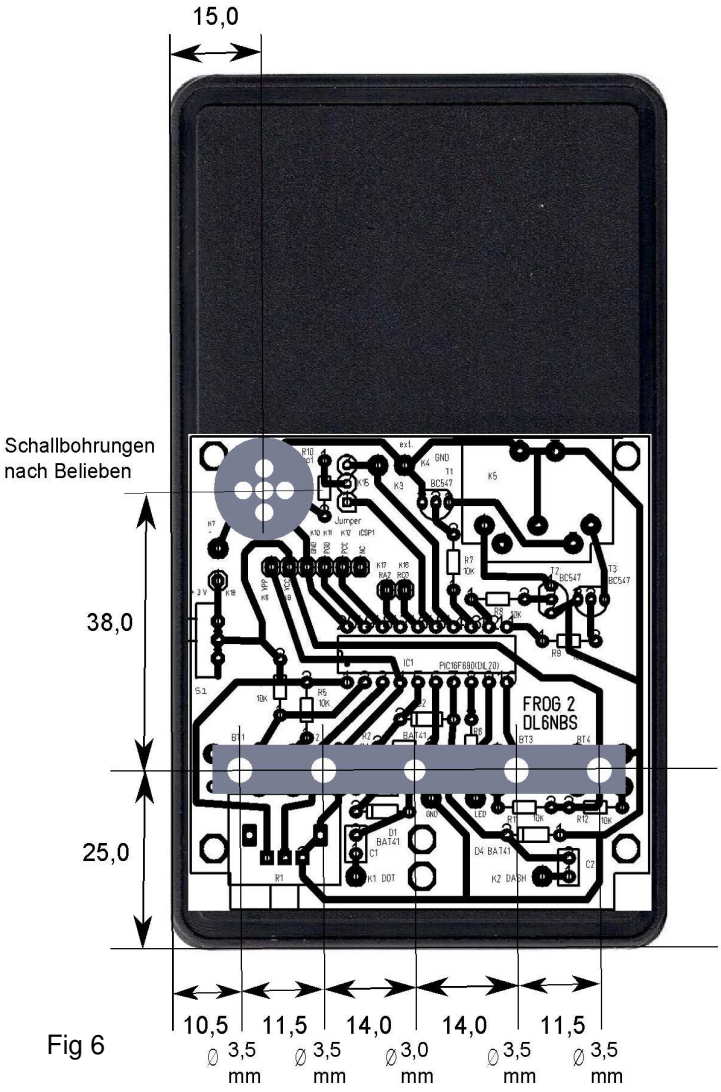



Fig 6

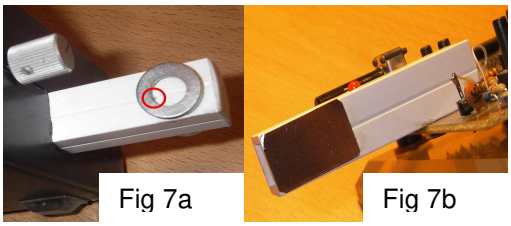


Fig 7a

Fig 7b

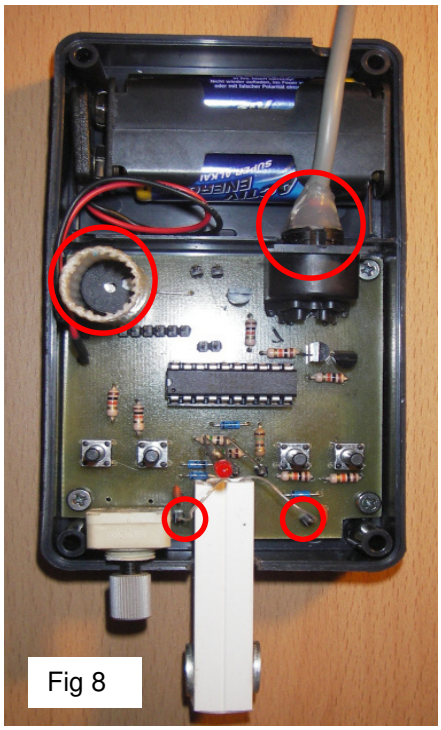


Fig 8

| FROG-2 by DL6NBS | | | | |
|---|---|---|---|---|
| X | X | SET | | ex |
| | | DEC | X | X |
| | X | TUN | X | |
| X | | WAK | | |
| | | RES | PWR + | X |

Fig 10

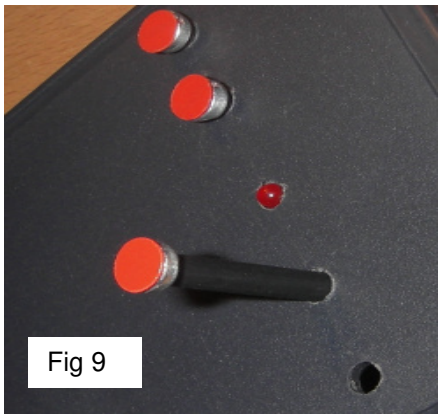


Fig 9

## 7. Feedback, suggestions, problems and resumé

Thanks for the testers feedback DJ9MH, DK1AX and DG8NCY.

Since version 2, the following bugs have been fixed and improvements implemented.
- Sleep mode starts after 60 minutes.
- Insecurity tune function eliminated.
- Output selectable contest number with 0 or T with leading T (3 digits).
- Bug in the counter decrement eliminated.
- Manual intelligent decrement buttons 3 +4 implemented.
- Decrement command removed.
- Seconds-inaccuracy in Rn command corrected.
- Entering "M" in the Setup causes the direct return to the operation mode.
- Squeeze- and Iambic- behavior improved.
- Command-sign is now : (colon) ---...
- Malfunction due to RF interference resolved by adding C3 to C6 and L1 to L3.

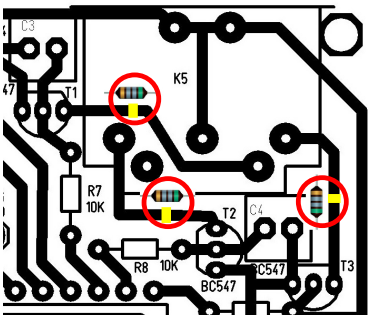



Fig 11

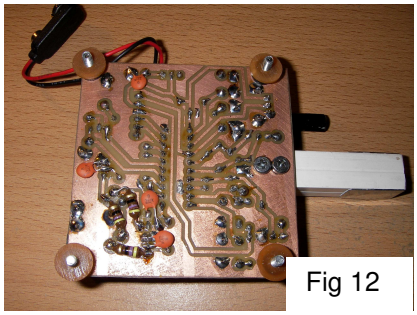


Fig 12

Cause of PIC's oscillator there two critical interfernces are produced near 30m and 40m Band. They can disturb weak RX-signals if the keyer is near the antenna (ex. MagLoop beside).