

# Project Buggalo

A project developed by SMØKBW - Bengt Andersson.

Buggalo is a combined bugg/keyer unit that has the ability to use audio-keying, I.E you can connect it to the microphone input on a SSB only station. You can use a strait key or paddles in IAMBIC style. You can use it as a traditional BUGG with ordinary keying output. There is 100 – 200 token memory capacity. The unit is operated only by two push buttons and the paddles.

This project is an open one, both hardware software. The software is placed under GPL license.

## History

The cause to the project was a purchase of a [MKARS80](#) kit, its a little QRP SSB station, Impossible to do any CW in its original shape.

The first plan was just to make a keyed audio oscillator with break-in (VOX). However as it evolved more and more functions has been implemented – and now it's a little bit like a Swiss army knife for CW operation.

Properties:

- Audio keyed by a strait key with break-in.
- Audio Bugg with break in.
- Traditional bugg with keyed output.
- One memory storage with the capacity of 100 – 200 token.
- Can be driven from the microphone bias.
- Audio monitor speaker when powered from an extern source.

I have chosen to put this project under GPL – Just as to thank to the HAM community for all it have given me in the past.

## Construction

The heart in the unit is a microcontroller – PIC12F693 alt. PIC12F675. Its Price is just around a US dollar. The PCB is through-hole and measures c.a 40x40 m.m. You can easily build it together with a small manipulator to get a small portable bugg.

Using the PIC12F683 gives a string memory of about 200 character space and the PIC12F675 will only give you around 100.

Buggalo can be driven from the microphone bias, but you have to modify the rig. The bias has to be on all the time – even in RX mode.

### connections:

**Conn1** – External power 5 – 14V:

1. Plus
2. Ground

**Conn2** – Rigg Connections:

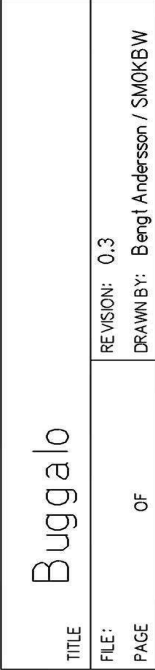
1. Audio out connect to microphone input. If you want to use the mic. bias, some mods has to be done.
2. PTT control / Bugg key output.
3. Ground
- 4.

**Conn3** – Monitor:

1. Speaker out
2. Ground
- 3.

**Conn4** – strait Key/manipulator:

1. Short paddle or strait key.
2. Long paddle or connect to ground when using a strait key.
3. Ground.



**Components**

R1	1,8 KΩ	C1	100 nF
R2	3,9 KΩ	C2	100 μF
R3	6,8 KΩ	C3	100 nF
R4	8,2 KΩ	C4	100 nF
R5	5,6 KΩ	C5	100 nF
R6	4,7 KΩ	C6	1 μF
R7	4,7 KΩ	C7	100 nF
R8	10 KΩ (potentiometer)	C8	10 nF
R9	470 Ω	C9	10 nF
R10	1 KΩ	C10	100 nF
R11	1 KΩ		
R12	100 KΩ		
R13	100 KΩ		
D1	1N4001	S1	Momentary connect
D2	Red 3mm LED	S2	Momentary connect
D3	Red 3mm LED		
D4	1N4148	The connectors can be of header type.	
T1	2N7000		
T2	2N7000		
U1	PIC12F683		

The PCB gives some space to different mounting of S1, S2 and R8.

## Function

Buggalo is controlled just by S1, S2 and the paddles. It's operated in IAMBIC mode A. I.E holding both paddles in gives a stream of "DI DA" or "DA DI" depending of witch was the first activated paddle.

The break-in functions works just like a VOX. The PTT line goes down when you is keying and continues being down for a little while after you stop keying.

Buggalo has a audio note of around 850 Hz. This makes it possible to work stations operating in CW mode. Most rigs are using USB in CW-mode.

You can choose to use strait key or the built in bugg in audio keying mode. Selection is done by conn2 - pin1, if that pin is grounded under reset (S1) the unit is in strait key mode. If bugg mode is selected you will hear a "B" in CW else it will be a "K" for Key.

The keying speed can be modified by press and holding either left or right paddle. Left will lower the speed and the right will increase the speed.

### Switches:

#### S1 Reset:

Restart the unit, if you for instance want to change keying mode.

#### S2:

Pressing only S2 will send the stored string.

#### S2 and left paddle:

Hold left paddle in at the same time as you press S2 will take you to Settings. All settings will be stored in EPROM and if saved even when the unit is turned off.

You will hear a "S" in CW, marking you are now in setting of the default speed. Left and right paddle adjust the speed.

To get to the next setting you only press S2. Doing so will you will hear a "U" for upper speed. This sets the max speed, when operating in normal mode.

Next press on S2 gives a "L" and you will set the lowest speed in normal mode.

Finally a press will give a "P" for pitch. You can adjust the pitch of the audio tone.

Next press on S2 will give you a "R" and the bugg is returned to normal mode.

You can break out from settings by pressing S1 – reset . Only completed steps will be stored I.E. the setting you are adjusting will be skipped and all following that. For example if you are in "U" settings and press reset, you will only have changed normal speed – "S".

**S2 and right paddle:**

Write string to memory.

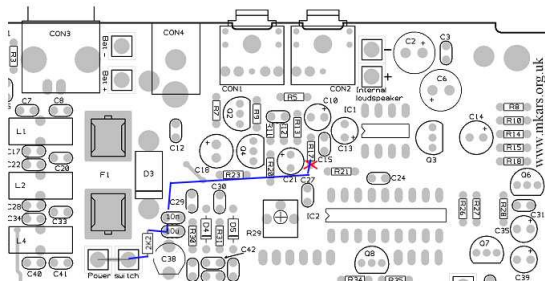
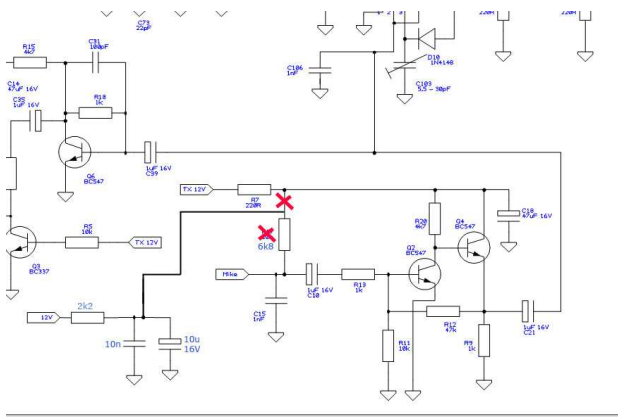
You are entering the letters by using “slap paddling”. A press on left marks a “DIT” and a right marks a “DAH”. Pressing both paddles marks end of letter and stores it in memory, this is marked by a longer beep. Word space is marked by by pressing both paddles one extra turn. A press on “S1” will stop entering and return the bugg in normal mode.

R8 is the level control of audio output to the rig.

## Modification of Mkars80

To use buggalo without external power some modifications has to be done to ensure that the mic bias is turned on even in RX.

1. Remove R17.
2. From the "Power switch" -hole nearest C38 solder a 2k7 resistor.
3. The other end of the resistor is decoupled by a 10nF and a 10uF capacitor to ground.
4. From that point mount a 6K8 resistor to the R17 hole that is nearest C10.



**Files:**

Buggalo.jpg	- Schematics.
buggalo.pdf	- PCB info
beskrivning.pdf	- Swedish description.
description.pdf this	- This document
COPYING.txt	- GPL info

In the src folder:

Buggalo.c  
Buggalo.h  
Eprom.h

Folder named 12F675\_hex contains the 12F675 hexfile.

Folder named 12F683\_hex contains the 12F683 hexfile.

**realization**

PCB and schematic is created in GEDA (Linux). Software is written in C and compiled in PICC / MPLAB. This document is created using PDFCreator.

OBS the dimensions in buggalo.pdf - PCB drawing is given in ISO standard. 1.000 is not one thousen, but 1 decimalpoint zero zero zero.

And last:

A big thank to my friends SMØOWG – Ove and SMØPVE – Ulf. Those guys have provided me with advice, support and have helped me with tests of the functionality of the bugg.

73 and good luck de  
SMØKBW / Bengt