A simple Roger Beep using a PICAXE microcontroller

Gaëtan Horlin, ON4KHG

---

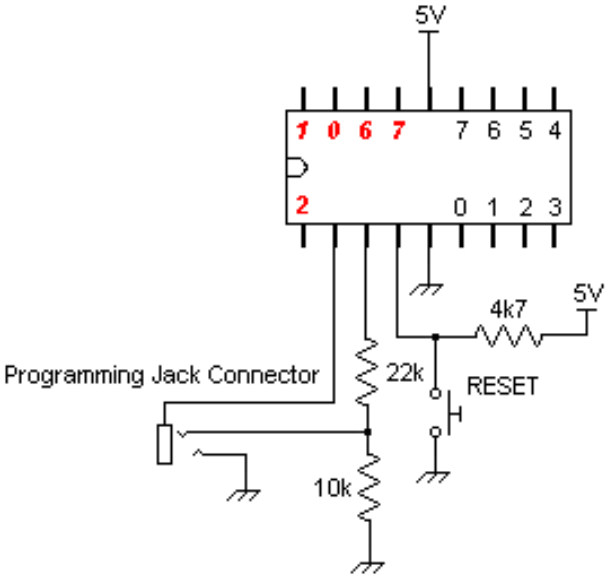
1. Introduction

Though it can be considered as a "gadget" or "CB inspired", a Roger Beep is quite useful when working small signals phone modes, prone to fading (QSB). It is also acting as a personal signature. The present paper describes a simple versatile Roger Beep capable of two different "melodies", a **K** (dah-dih-dah) and a **Bell Tone**. It makes use of a microcontroller that can be programmed in situ with a RS232 or USB cable connected to a computer.

2. Presentation of the PICAXE microcontroller

The PICAXE is a Microchip microcontroller pre-programmed with a bootstrap program that enables a direct cable download. Blank microcontrollers don't contain this bootstrap program and so can't be programmed from within the PICAXE system which uses a very simple interface (3 wires connection) to the computer serial (RS232) or USB port. Although this interface doesn't use true RS232 voltages, it is very low-cost and has proven to work reliably on almost all modern computers. As the PICAXE flash memory can be programmed in situ, there is no need for an external programmer, so avoiding to damage the IC when plugging and un-plugging it.
The PICAXE can be programmed with a set of BASIC instructions or even with flowcharts, through the Programming Editor software.
The PICAXE and accessory components (programming cable, tutorial boards,…) are supplied by Revolution Education Ltd (http://www.rev-ed.co.uk) in United Kingdom, at cheap prices.
All the necessary information about the PICAXE's (pinning, set of instructions, interfacing,…) is provided on the same website through three user guides (Manuals 1, 2 & 3).
The Programming Editor software runs fine on Windows© XP but I have not tested it on Vista.

The present project makes use of the **PICAXE-18** of which minimum functional schematic is given below :



0, 1,…, 7 = Outputs
0, 1, 2, 6, 7 = Inputs

(***0, 1, 2*** are digital or analogue Inputs)

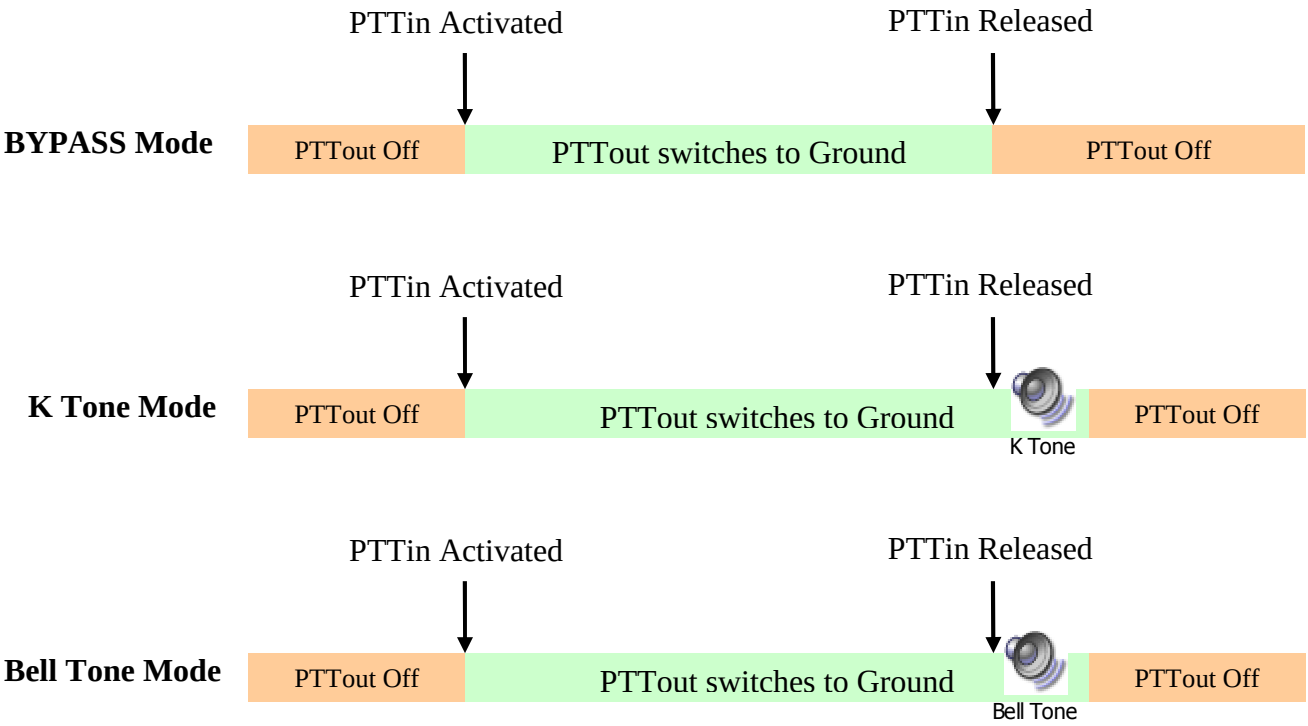
These are Input/Output numbers,
not physical IC pin numbers (1-18)

Pins 2 & 3 are used for the flash memory programming of the IC, via the serial cable to the RS232 port of the computer. Pin 4 is the RESET.

3. The Hardware

3.1. Principle of working

The working of the Roger Beep is depicted on the following chronogram :



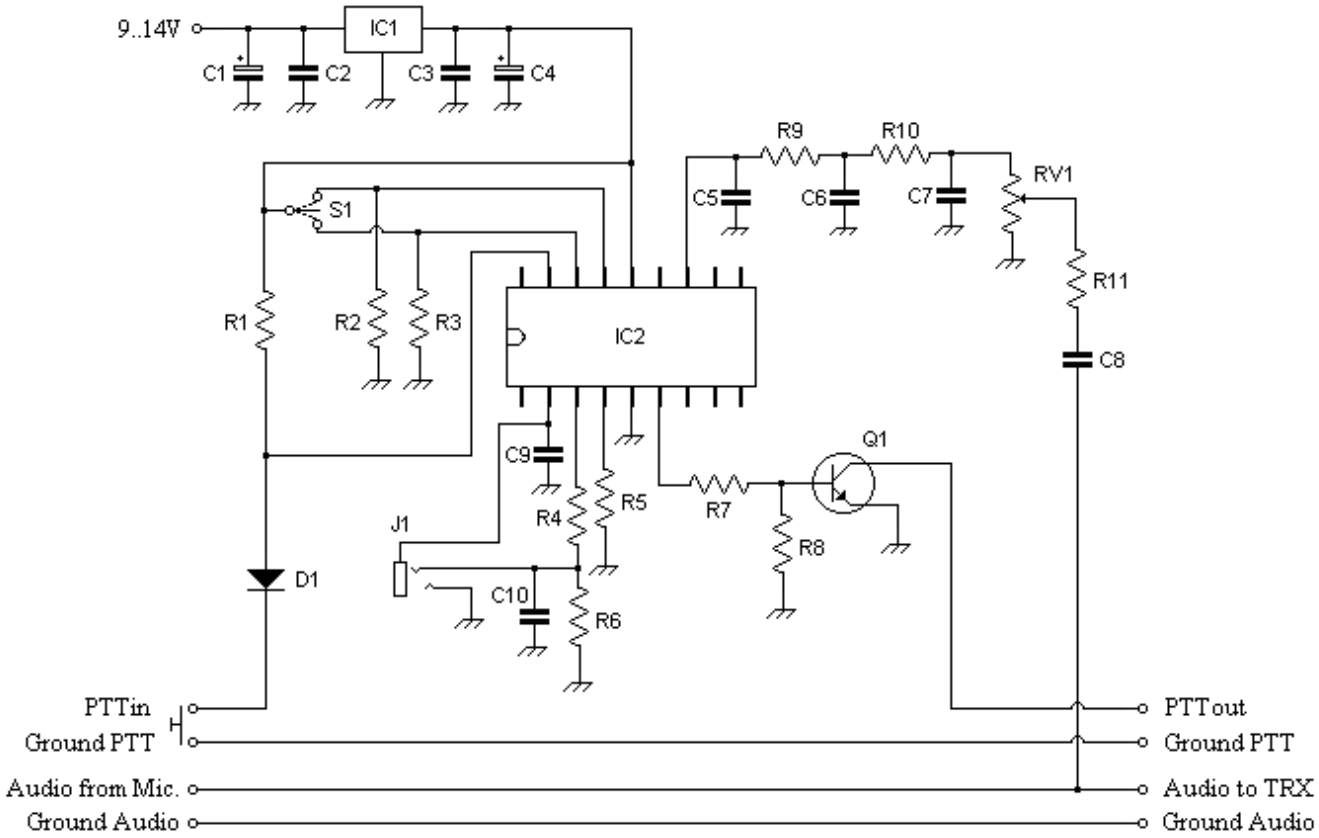
Clic on the headphones to listen to how the tones sound like.

The PTTin is ground activated but can easily be modified for a high level activation (the software program must also be modified accordingly); the PTTout is delivered on an open collector (shorted to ground for subsequent transceiver activation in TX mode).

Here, the RESET button between pin 4 and the ground has been knowingly omitted.

3.2. Electronic Schematic

As we deal here with low frequencies, the PCB layout is not critical. Hence, the circuit has been built on a Veroboard.



The switch S1 allows to select one of the three operating modes.
The audio tones out of the PICAXE are filtered through R9, R10, C5, C6 & C7 and their amplitude is adjusted by RV1 and R11 (make sure not to saturate the microphone input of your transceiver by a too high beep volume).
IC1 supplies the circuit with the 5V required. Never supply the PICAXE with more than 6V.

3.3. Part List

Part ID	Value	Part ID	Value
R1, R2, R3, R6	10kΩ	C4	1µF
R4	22kΩ	C9, C10	10nF
R5, R7, R8	4,7kΩ	D1	1N4148
R9, R10	1,2kΩ	Q1	2N2222
R11	100kΩ	IC1	78L05
RV1	100kΩ Linear	IC2	PICAXE-18
C1	2,2µF	S1	ON-OFF-ON Switch
C2, C3, C5, C6, C7, C8	100nF	J1	3,5 mm Stereo Jack

4. The Software

The program is written so that when none of the two available Roger beep tones is selected (through S1), the circuit is set in Bypass Mode, i.e. the PTTin activation to ground is exactly reflected on the PTTout port.

The program is given below. The text in green (and starting by ') is not part of the set of instructions, it is just an explicative text to ease the understanding of the program ; it will be ignored by the Programming Editor when downloading the program into the PICAXE.

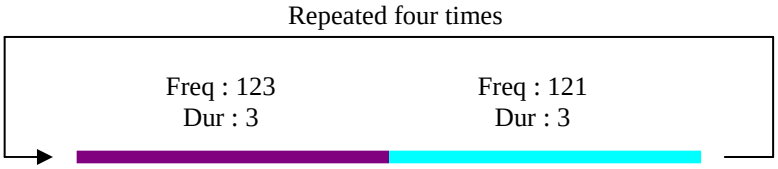
```
'This program is to be used with a PICAXE-18 to generate end of transmission tones.
'Copyright Gaëtan Horlin ON4KHG - March 2009

MAIN:                                       'Label of the MAIN sub-program
        if pin6 = 0 and pin7 = 0 then BYPASS 'If Input 6 and Input 7 are at low level, jump to BYPASS
        if pin0 = 0 then PTT_ON             'If Input 0 is at low level  (=PTTin pressed), jump to PTT_ON
        goto MAIN                           'As long as conditions above are not met, go back to MAIN

PTT_ON:                                     'Label of the PTT_ON sub-program
        outpin0 = 1                         'Set Output 0 to a high level (=PTTout shorted to ground)
        if pin0 = 1 and pin6 = 1 then BELL_TONE 'If Input 0 and Input 6 at high level, jump to BELL_TONE
        if pin0 = 1 and pin7 = 1 then K_TONE 'If Input 0 and Input 7 at high level, jump to K_TONE
        goto PTT_ON                         'As long as conditions above are not met, go back to PTT_ON

BELL_TONE:                                  'Label of the BELL_TONE sub-program
        for b0 = 1 to 4                     'Set variable b0 vary from 1 to 4 (*)
        sound 6,(123,3,121,3)               'Generate the bell tone on Output 6. See text for more details
        next b0                             'Increment b0 and loop back to line (*) until b0 = 4
        pause 50                            'Pause 50 ms
        outpin0 = 0                         'Set Output 0 to a low level (=deactivate PTTout)
        goto MAIN                           'Go back to MAIN

K_TONE:                                     'Label of the K_TONE sub-program
        sound 6,(119,12,0,5,119,4,0,5,119,12) 'Generate the bell tone on Output 6. See text for more details
        pause 50                            'Pause 50 ms
        outpin0 = 0                         'Set Output 0 to a low level (=deactivate PTTout)
        goto MAIN                           'Go back to MAIN

BYPASS:                                     'Label of the BYPASS sub-program
        outpin0 = 1-pin0                    'Set Output 0 to the inverse level of Input 0
        goto MAIN                           'Go back to MAIN
```

The **Sound** instruction has the following syntax : SOUND #outpin,(note,duration,note,duration,…).

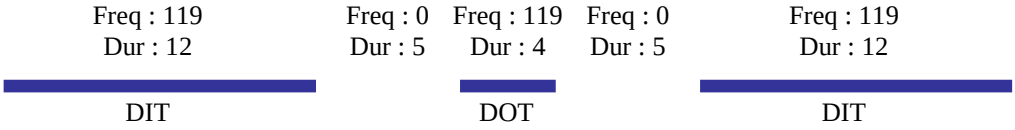
- **#outpin** is the number of the output onto which the tones are delivered [0-7]
- **note** specifies the type and frequency of the tones [0-255]
  - [0-127] produce frequency ascending tones
  - [128-255] produce frequency ascending white noises
  - [0] produces a silence
- **duration** defines the length of the tones in multiples of 10 ms [0-255]

The Bell Tone is made of looping four times two tones of different frequencies, each of a 30 ms duration.

The Bell Tone structure is :



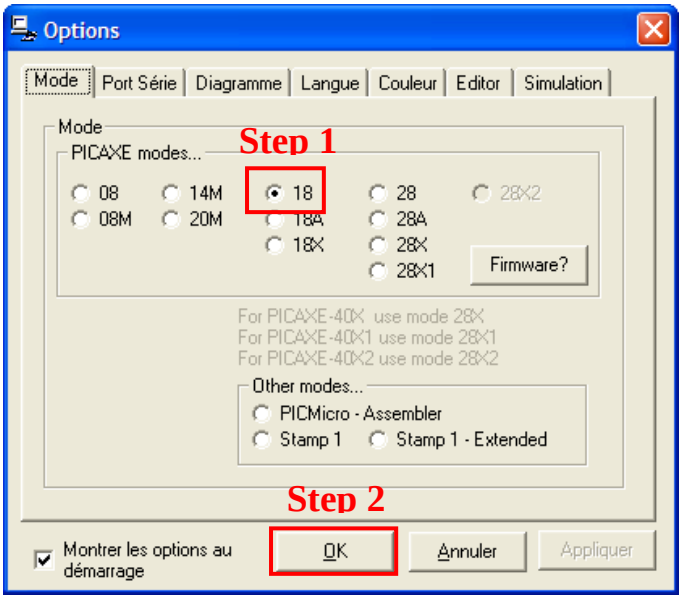
And the K Tone :

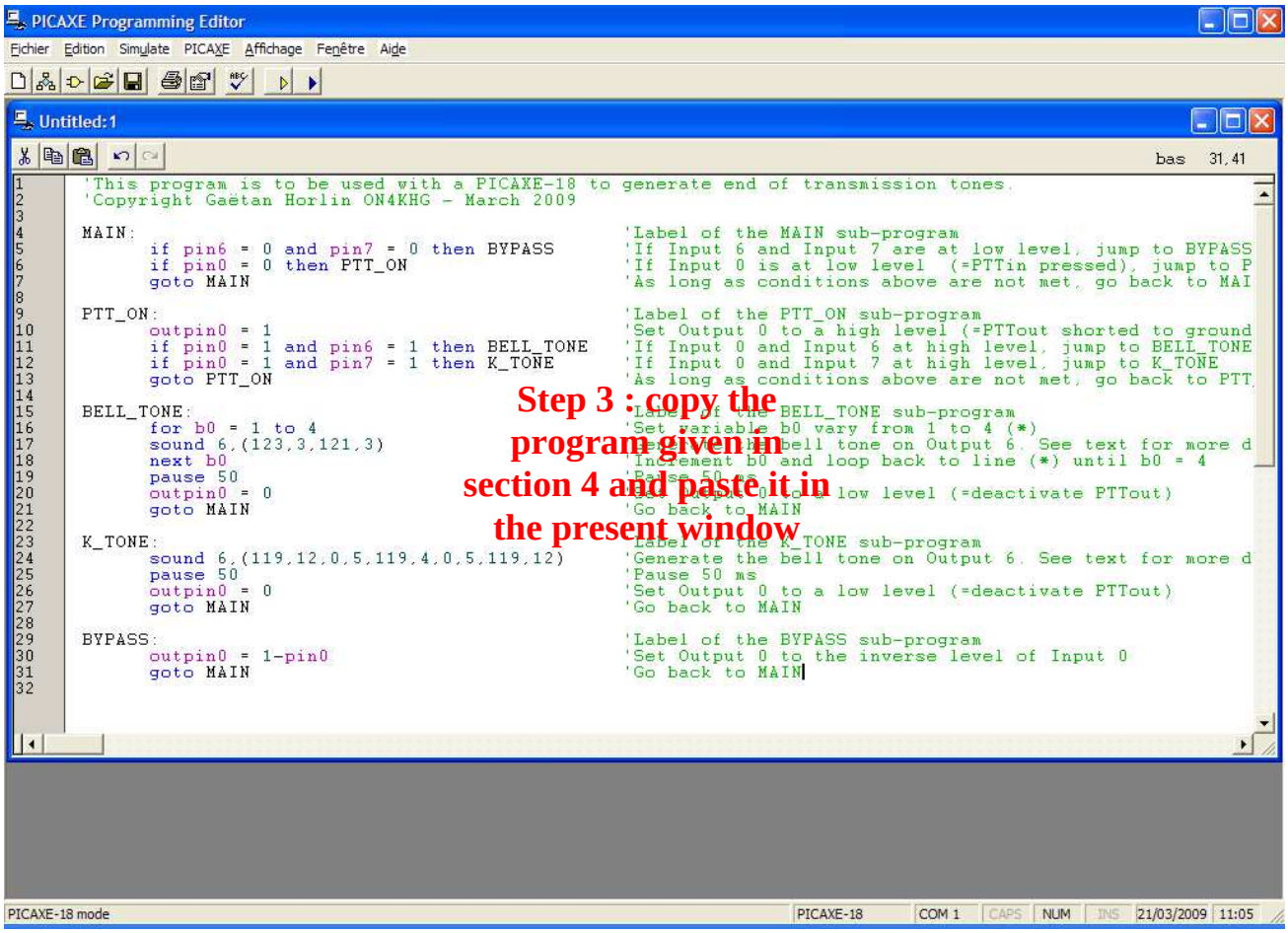


5. Programming the PICAXE

It is quite straightforward to program the PICAXE.
First connect the programming connector (J1) of the Roger Beep circuit to the RS232 (or USB through a special adaptor) port of the computer through the programming cable available from Revolution Education Ltd (http://www.rev-ed.co.uk).

Then, open the PICAXE Programming Editor software.
You get the windows below; just follow the steps.





The final operation (Step 4) is to download the program into the PICAXE :

That's all !

6. Document History

Date	Device Version	Comment
21/03/2009	1st	Creation of the document