ZL1BPU LF Exciter

Operator's Reference Guide

M. Greenman CSc, ZL1BPU
September 2002

Summary of Maths Used

Frequency resolution

The frequency resolution is related to the size of the phase accumulator register and the crystal or TCXO **Reference Frequency**.
Resolution = Reference Frequency (Hz) / (9 x 2^{24}) Hz …[1]

With a 12.8 MHz crystal frequency, this works out at 0.0847710503472 Hz.

Operating Frequency

To calculate the value to send to the Exciter for any given frequency, divide the frequency by the frequency **Resolution**, with all the many decimal places.

Frequency Value = Reference Frequency / Resolution …[2]

Commands must always be 24 bit (six hex characters).

For example, for 181400 Hz, 181400 / 0.0847710503472 = 2139881.472000 (round to nearest integer).

Then convert to hexadecimal. For example 2139881_{10} = $20A6E9_{HEX}$

Offset

The frequency offset can be calculated by dividing the offset in Hz by the Resolution. Use equation (2), and then convert to hex. Commands must always be 8 bit (two hex characters). The maximum value is 255_{10} (FF_{HEX}).

Keying Rate

The rate at which symbols occur is called the *baud rate*, and is dependent on the external symbol clock frequency. This can be from 50 to 500 Hz, but must be known with at least 1% accuracy.

To define the required baud rate, you have to specify the division ratio for the micro controller timer.

Timer Value = Symbol Clock (Hz) / baud rate …[3]

Then convert the value to hexadecimal. The value must always be 16 bit (four hex characters).

The baud rate for Morse is 1.2 x the speed required in WPM.

For modes where you need to express the baud rate in “so many second dots” (dot period) as for QRSS, the formula becomes:

Timer Value = Symbol Clock (Hz) x dot period …[4]

As an example, lets say we have a 64 Hz symbol clock. To generate 3 second dots, use a timer value of 3 x 64 = 192 ($00C0_{HEX}$). The values for the **K** command, which sets the keying rate, are always expressed as four characters. So, we send the command “**K00C0**”.

Sweep Width

A hexadecimal 8 bit number (00 to FF), representing the number of steps.

Sweep Frequency Increment

As for operating frequency, but limited to a 16 bit number (four hex characters).

Sweep Dwell Time

Timing and resolution depend on the main crystal or TCXO Reference Frequency. The timer is prescaled from the clock x 1024, so the timer resolution is given by:

Timer Resolution (ms) = 1024 / Reference Frequency (kHz) …[5]

The time per step setting for the **Dwell Timer** is calculated by dividing the time required by the **Timer Resolution**:

Dwell Timer = Time (ms) / Timer Resolution (ms) …[6]

Resolution is about 12 milliseconds and range is well under 1ms (1ms = A0C) to over 20ms (20ms = AF8). The value is converted to hexadecimal. It is limited to an 8 bit number (00 to FF).

Serial Commands (KISS commands)

Axx ADD
Add offset of xx resolution steps, where xx is "00" to "FF". In Sweep Generator mode, sets the sweep dwell time (time per step). Resolution is about 1/12 milliseconds and range is under 1ms (1ms = A0C) to over 20ms (20ms = AF8).

B BEACON
Command to enter beacon text, where up to 120 bytes of data and commands can be entered. The data consists of HEX-ASCII character pairs. <CR> and <SPACE> are permitted, and the last character must be "FF". Data entry mode is terminated by the tilde character " ~ ". The message is permanently stored and the unit is reset on exit.

Fhhmmll FREQUENCY
Set frequency to this value times the resolution. The value hhmmll is a 24 bit binary number expressed as six hexadecimal characters, "00" to "FF".

H HELP
Simple help message listing these commands (also resets microcontroller).

Knnnn KEY
The beacon mode keying baud rate, where nnnn is 0x0000 to 0xFFFF. Resolution is 31.25ms, but is dependent on the symbol clock generator frequency (typically 32 Hz). Hence K0001 gives a baud rate of 32 baud - 3 sec dot QRSS requires K00C0 and JASON (11.8 sec symbols) K02F0. In Sweep Generator mode, sets the frequency increment in resolution steps (as for the F command), but range is limited to K0000 to KFFFF (over 5 kHz/step).

Mn MODE
Sets the beacon mode, where n is a number 0 to 6.

0 **Beacon Off** Continuous carrier controlled by A, F, T, W and X commands
1 **ASK** On-Off single frequency keying, beacon message in Morse
2 **FSK** Continuous carrier FSK keying, beacon message in Morse
3 **DFSK** Dual frequency on/off keying, beacon message in Morse
4 **MFSK** Eight frequency bit-mapped scanned MFSK mode (e.g. MT-Hell)
5 **HELL** On-Off single frequency bit-mapped keying (e.g. Feld-Hell)
6 **MFSK/IFK** MFSK data frequency shift in resolution units (e.g. JASON)

Pp PORT
Sets Port D outputs PD2, PD3, PD4 according to value least significant bits of value "p" as follows:

0	PD2=0 PD3=0 PD4=0	4	PD2=0 PD3=0 PD4=1
1	PD2=1 PD3=0 PD4=0	5	PD2=1 PD3=0 PD4=1
2	PD2=0 PD3=1 PD4=0	6	PD2=0 PD3=1 PD4=1
3	PD2=1 PD3=1 PD4=0	7	PD2=1 PD3=1 PD4=1

R REPORT
Requests a message giving the current settings. Reports A K M W and F.

S STORE
Store current settings. Saves FREQUENCY, OFFSET, MODE and KEY. Sweep mode settings are not saved.

T TX
T turns the transmitter on.

Wmm WIDTH
Sets number of sweep steps, where mm is "00" to "FF" in two hexadecimal characters. Typical value is W14 for 20 steps. 50 baud 170 shift RTTY with 2125 Hz MARK can be simulated with W02 AF8 K07E5 F0062B1. The start frequency for the sweep is set by the F command, sweep dwell time by the A command, and the sweep step size by the K command. W00 turns off the sweep.

X RX
X turns the transmitter off.

Beacon (script) Commands

F1	Mode 1 ASK Morse
F2	Mode 2 FSK Morse
F3	Mode 3 DFSK Morse
F4	Mode 4 MFSK Graphics (Sequential MT-Hell)
F5	Mode 5 ASK Graphics (Feld-Hell)
F6	Mode 6 MFSK/IFK Data (Jason)
FB pp	Set port pins (see **P** command)
FC hhmmll	Set frequency (see **F** command)
FD nn	Set FSK shift or MFSK increment (see **A** command)
FE nn nn	Set keying speed (baud rate - see **K** command)
FF	End of script

The FB, FC, FD and FE commands are functional replicas of the corresponding KISS commands.

Morse Coding

Using “Murphy” coding, each dot is represented by “0” and each dash by “1” packed right-to-left in a single byte. A “1” is placed in the next free bit to the left to signify end of character, and remaining bits to the left remain “0”. As each element of the character is transmitted, the data byte is shifted right and the carry transmitted (and followed by inter-element spaces) until the remaining data is 0x01 (i.e. “00000001”).

Coding is arranged so that automatic inter-character spaces are correct at three dot-lengths, and since “space” is coded as no data elements, but followed by an inter-character space, the inter-word space is six dot-times.

Some characters are interpreted as inter-character spaces (0x01), while others are ignored completely (0x00). The table below is represented in hexadecimal. Lower case characters should be coded as upper case.

SP	01	0	3F	A	06	Q	1B
!	00	1	3E	B	11	R	0A
“	52	2	3C	C	15	S	08
#	00	3	38	D	09	T	03
$	C8	4	30	E	02	U	0C
%	01	5	20	F	14	V	18
&	01	6	21	G	08	W	0E
‘	5E	7	23	H	10	X	19
(	2D	8	27	I	04	Y	1D
)	6D	9	2F	J	1E	Z	13
*	01	:	47	K	0D	[	00
+ AR	2A	;	35	L	12	\ SK	68
,	73	<	00	M	07	]	00
-	61	= BT	31	N	05	^	00
.	6A	>	00	O	0F	_	6C
/	29	?	4C	P	16	`	00

Hell Font

The suggested font is a 5x7 low resolution Hell font. Characters are listed in ASCII order, starting at ASCII 32 (space). Five bytes are shown for each character, represented in HEX. Trailing spaces can be ignored, making the font proportional. Always include at least one byte of 0x00 (zero) between characters to act as a character space. For double-width characters, repeat each byte - for Feld-Hell, four or more repeats.

SP	00,00,00,00,00	0	0E,13,15,19,0E
!	00,1D,00,00,00	1	00,10,1F,00,00
“	00,18,00,18,00	2	03,15,15,15,09
#	0A,1F,0A,1F,0A	3	15,15,15,0A,00
$	08,15,1F,15,02	4	1E, 02,07,02,00
%	19,1A,04,0B,13	5	1C,15,15,15,02
&	0A,15,0D,02,05	6	0E,15,15,15,02
‘	00,10,18,00,00	7	10,10,17,18,00
(	00,0E,11,00,00	8	0A,15,15,15,0A
)	00,11,0E,00,00	9	08,15,15,15,0E
*	15,0E,1F,0E,15	:	00,05,05,00,00
+	04,04,1F,04,04	;	00,0A,0B,00,00
,	00,01,06,00,00	<	04,0A,11,00,00
-	04,04,04,04,04	=	00,0A,0A,0A,00
.	00,03,03,00,00	>	00,11,0A,04,00
/	01,02,04,08,10	?	08,10,15,14,08
@	0E,11,15,15,08	P	1F,14,14,14,08
A.	0F,14,14,14,0F	Q	1F,11,15,13,0F
B	1F,15,15,15,0A	R	1F,14,16,15,09
C	0E,11,11,11,00	S	08,15,15,15,02
D	1F,11,11,11,0E	T	10,10,1F,10,10
E	1F,15,15,15,11	U	1F,01,01,01,1F
F	1F,14,14,14,10	V	18,06,01,06,18
G	0E,11,11,13,0A	W	1E,01,06,01,1E
H	1F,04,04,04,1F	X	11,0A,04,0A,11
I	00,11,1F,11,00	Y	10,08,07,08,10
J	02,11,1F,10,00	Z	11,13,15,19,11
K	1F,04,0A,11,00	[	1F,11,11,00,00
L	1F,01,01,01,00	\	10,08,04,02,01
M	1F,08,04,08,1F	]	11,11,1F,00,00
N	1F,08,04,02,1F	^	00,08,10,08,00
O	1F,11,11,11,1F	_	01,01,01,01,01

Contents of EEPROM

The micro controller EEPROM contains a number of default values and the beacon script. For those who prefer to change the values or load messages directly, here is a list of the contents, in address order.

Address (HEX)	Parameter / Contents	Comment
0000 - 0001	0x00	Unused
0002 - 0004	USRHI, USRMD, USRLO	Default operating frequency (Fhhmmll cmd)
0005	MODE	Default operating mode (Mn command)
0006	OFFSET	Default FSK offset (Amm command)
0007	UART	UART (link to PC) data rate
0008 - 000F	0xFF	Reserved
0010	-	First byte of user beacon message
? (10 < ? < 7F)	0xFF	End of user message
? to 0x7F	0xFF	Unused

The contents of the first 16 bytes, if the supplied .eep file is used, will be:

00 00 20 E8 33 00 18 52 Default parameters
00 30 00 02 FF FF FF FF Reserved and unused

A typical test message (in HEX) follows. If this message is loaded in the order given from address 0x0010, "DE ZL1BPU ZL1BPU AR" will be sent in Morse at 10 WPM, on 181.400 kHz (with 12.8 MHz reference).

FC 20 A6 E9	Set frequency 20A6E9 (181.4 kHz)
FB 01	Set port PD2 on
F1	Set ASK mode M1
FE 00 04	Set speed K0004 (about 10 WPM)
01 09 02 01	(sp)DE(sp)
01 13 12 3E 11 16 0C 01	(sp)ZL1BPU(sp)
01 13 12 3E 11 16 0C 01	(sp)ZL1BPU(sp)
11 15 05 2A	BCN(AR)
FF	End of message