Kronos Program Bank Expansion Proposal

Tim Godfrey (V1.1 2-Jan-2015)

The need for additional program banks for the Kronos has been discussed and requested since the introduction of the instrument. Most owners want to keep the all the factory installed programs in their standard locations, so only the seven expanded user banks U-AA to U-GG are available. The fixed nature of HD-1 and EXi banks further reduces the efficiency of use. As Kronos owners continue to purchase, download, and create additional programs and banks, managing the limited space within the Kronos user program banks becomes a significant chore.

Excellent software like PCG Tools helps a lot with the management aspect, but it still has to be done. Being able to audition programs directly from a PCG file on disk is helpful also, but doesn't eliminate the need for more program banks.

The first step is to come to terms that additional banks will not be able to be directly accessible from front panel buttons with unique LED patterns. Korg did a good job coming up with the U-AA to U-GG "double button" scheme, but that approach can only go so far.

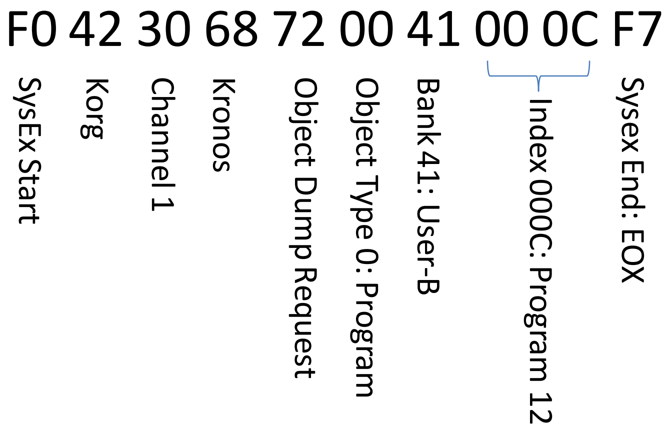
Most users would be happy to have the additional banks available, even if they could only be accessed through the touchscreen interface. If the lack of direct button access to the program banks is a problem for any user, they don't have to use the extra banks. The current bank system is not changed. The Bank Program Select screen already has "More" tabs at the bottom, so there is no hard limit to how many "more" pages of tabs could be added – only the practicality of moving through the pages.

The limit for number of banks in the Kronos ultimately comes from the limits of MIDI. Internally, bank numbers are represented by a byte, giving a total of 256 possible values. When the Bank number is transmitted as a part of a System Exclusive message, it appears in the message header without any byte encoding. This means the value can use only 7 bits (the MSB must be 0). Therefore the range is 0 to 127.

Background on MIDI SysEx:

Program banks and other data structures are communicated over MIDI using System Exclusive messages. Korg defines the SysEx message formats in a document titled "KRONOS_MIDI_SysEx.txt", which is available at the Korg support site under Operating System, MIDI System Exclusive Documentation.

SysEx messages are vendor-specific by design. Each message starts with a header that includes a unique ID for the vendor, and typically an identifier for a particular instrument. A SysEx header for the Korg Kronos looks like this: F0 42 3g 68 (all numbers are in hexadecimal). F0 is the SysEx start byte, 42 is Korg's vendor ID 30 is the MIDI channel (often listed as 3g, where 'g' is replaced with the Midi General channel for the instrument, minus 1 ---- Channel 1 is coded as 30. Channel 2 as 31, etc). 68 is the ID for the Kronos. This is followed by a "Function" code, which specifies what the function of the message is (such as bank dump, dump request, parameter change, etc). For bank dumps or requests, the function is followed by the Object type (whether it is a Program, Combi, Setlist, etc), and then the Bank ID. This is followed by an index which specifies which program, combi, or setlist we are referencing within the bank. It uses 2 bytes because some structures like a setlist are larger than 128. If the message is a dump request, it ends here with the End of Exclusive (EOX) character F7. If it is a dump, the data is sent, followed by the EOX. Here is an example dump request for Kronos Program 012 in Bank User-B.



F0 42 30 68 72 00 41 00 0C F7

The important thing to understand is that this message format is very specific. By including codes to specify Korg, Kronos, and Object Type, we are able to unambiguously reference 128 unique program banks for the Kronos. This can be done without changing any message formats (only the allowed values used in the bank byte are changed. The length of messages and the meaning of all the bytes remain the same.

Table 1 below shows the current (as of OS V2.1 MIDI implementation documentations – which is apparently not changed for OS V3). Program banks use the values from 0 to 0x4D (77 decimal), but include gaps in that range.

Table 1 - System Exclusive “Bank” definitions[1]

```
[73] Object Dump                      Receive/Transmit
        F0, 42, 3g, 68          Excl Header
        73                      Function
        obj                     *1
        bank                    *2
        idH                     Index (bit7-13)
        idL                     Index (bit0-6)
        version                 obj's version number
        data                    *3
        F7                      End of Excl


*1
        obj = 00 : Program (Prog_EXi_Common.txt, Prog_EXi.txt, Prog_HD-1.txt)
              01 : Combination (CombiAndSongTimbreSet.txt)
              02 : Song Timbre Set (CombiAndSongTimbreSet.txt)
              03 : Global (Global.txt)
              04 : Drum Kit (DrumKit.txt)
              05 : Wave Seq (WaveSequence.txt)
              06 : KARMA GE (NOTE: GEs are not edited on the instrument, so
                   there is no edit buffer)
              07 : KARMA Template
              08 : Song Control (SongControl.txt)
              09 : Song Event (currently disabled)
              0A : Song Region
              0B : Reserved
              0C : KARMA GE RTP Info (KARMA_GE_RTP.txt)
              0D : Set List (index=set list) (SetList.txt)
              0E : Drum Track Pattern (DrumTrackPattern.txt)
              0F : Drum Track Pattern Event (DrumTrackPatternEvent.txt)
              10 : Set List Slot Comments (bank=set list, index=slot)
              11 : Set List Slot Name (bank=set list, index=slot)
              12 : Combi Name
              13 : Program Name
              14 : Song Name
              15 : Wave Seq Name
              16 : Drum Kit Name
              17 : Set List Name (index=set list)
              18 : Song (Song Timbre Set and Song Control in a single object. Song.txt)

 The table above includes the names of associated documentation files in SysExDumps.


*2
 The meaning of bank depends on the object type.

 Program, Program Name:
        bank =  0 -  5 : INT-A - F
               10 - 1A : GM, g(1)-g(9), g(d) (read-only)
               40 - 4d : USER-A - G, AA - GG

 Combi, Combi Name:
        bank =  0 -  6 : INT-A - G
               40 - 46 : USER-A - G

 Drum Kit, Drum Kit Name:
        bank =       0 : INT
                    10 : GM (read-only)
               40 - 4d : USER-A - G, AA - GG

 Wave Seq, Wave Seq Name:
        bank =       0 : INT
               40 - 4d : USER-A - G, AA - GG

 KARMA GE:
        bank =  0 -  B : USER-A - L

 KARMA Template:
        bank =  0 -  3 : USER-A - D

 KARMA GE RTP Info:
        bank =  0
        index = 0 - 2047: Preset
                2048 -  : U-A000 -

 Set List Slot Comments, Set List Slot Name
        bank =  0 - 7F : set list #

 For all other types (e.g., Song, Set List, etc) bank must be 0.
```

[1] From “KRONOS_MIDI_SysEx.txt” for MIDI Implementation OS V2.1

Table 2 below shows two proposals for expanding the program banks. The general idea is to continue the two-letter User bank style started with U-AA, but using U-XA to U-XZ, then U-YA and so on.
The simple proposal is presented in column D and simply continues the assignment of program banks beyond U-GG. This is exactly what was done by Korg when the banks U-AA through U-GG were added. This approach adds 50 new banks, bringing the total usable banks to 70, which is a tremendous improvement over the 20 now available.

Instead of adding only 50, all 97 undefined banks could be used. That would give a total of 117 usable program banks. This proposal is presented in column E. It starts with U-WA and runs through U-ZU.

Table 2 - Program Bank Expansion Options

Decimal	Hex	Current Use	Proposed (Basic Option)	Proposed (Full Option)
0	0	INT-A	INT-A	INT-A
1	1	INT-B	INT-B	INT-B
2	2	INT-C	INT-C	INT-C
3	3	INT-D	INT-D	INT-D
4	4	INT-E	INT-E	INT-E
5	5	INT-F	INT-F	INT-F
6	6			U-WA
7	7			U-WB
8	8			U-WC
9	9			U-WD
10	A			U-WE
11	B			U-WF
12	C			U-WG
13	D			U-WH
14	E			U-WI
15	F			U-WJ
16	10	INT-G GM G(1)	INT-G GM G(1)	INT-G GM G(1)
17	11	INT-G GM G(2)	INT-G GM G(2)	INT-G GM G(2)
18	12	INT-G GM G(3)	INT-G GM G(3)	INT-G GM G(3)
19	13	INT-G GM G(4)	INT-G GM G(4)	INT-G GM G(4)
20	14	INT-G GM G(5)	INT-G GM G(5)	INT-G GM G(5)
21	15	INT-G GM G(6)	INT-G GM G(6)	INT-G GM G(6)
22	16	INT-G GM G(7)	INT-G GM G(7)	INT-G GM G(7)
23	17	INT-G GM G(8)	INT-G GM G(8)	INT-G GM G(8)
24	18	INT-G GM G(9)	INT-G GM G(9)	INT-G GM G(9)
25	19	INT-G GM G(A)	INT-G GM G(A)	INT-G GM G(A)
26	1A	INT-G GM G(D)	INT-G GM G(D)	INT-G GM G(D)
27	1B			U-WK
28	1C			U-WL
29	1D			U-WM
30	1E			U-WN
31	1F			U-WO
32	20			U-WP
33	21			U-WQ
34	22			U-WR
35	23			U-WS
36	24			U-WT
37	25			U-WU
38	26			U-WV
39	27			U-WW
40	28			U-WX
41	29			U-WY
42	2A			U-WZ
43	2B			U-XA
44	2C			U-XB
45	2D			U-XC
46	2E			U-XD
47	2F			U-XE
48	30			U-XF
49	31			U-XG
50	32			U-XH

Decimal	Hex	Current Use	Proposed (Basic Option)	Proposed (Full Option)
51	33			U-XI
52	34			U-XJ
53	35			U-XK
54	36			U-XL
55	37			U-XM
56	38			U-XN
57	39			U-XO
58	3A			U-XP
59	3B			U-XQ
60	3C			U-XR
61	3D			U-XS
62	3E			U-XT
63	3F			U-XU
64	40	U-A	U-A	U-A
65	41	U-B	U-B	U-B
66	42	U-C	U-C	U-C
67	43	U-D	U-D	U-D
68	44	U-E	U-E	U-E
69	45	U-F	U-F	U-F
70	46	U-G	U-G	U-G
71	47	U-AA	U-AA	U-AA
72	48	U-BB	U-BB	U-BB
73	49	U-CC	U-CC	U-CC
74	4A	U-DD	U-DD	U-DD
75	4B	U-EE	U-EE	U-EE
76	4C	U-FF	U-FF	U-FF
77	4D	U-GG	U-GG	U-GG
78	4E		U-XA	U-XV
79	4F		U-XB	U-XW
80	50		U-XC	U-XX
81	51		U-XD	U-YA
82	52		U-XE	U-YB
83	53		U-XF	U-YC
84	54		U-XG	U-YD
85	55		U-XH	U-YE
86	56		U-XI	U-YF
87	57		U-XJ	U-YG
88	58		U-XK	U-YH
89	59		U-XL	U-YI
90	5A		U-XM	U-YJ
91	5B		U-XN	U-YK
92	5C		U-XO	U-YL
93	5D		U-XP	U-YM
94	5E		U-XQ	U-YN
95	5F		U-XR	U-YO
96	60		U-XS	U-YP
97	61		U-XT	U-YQ
98	62		U-XU	U-YR
99	63		U-XV	U-YS
100	64		U-XW	U-YT
101	65		U-XX	U-YU
102	66		U-XY	U-YV
103	67		U-XZ	U-YW
104	68		U-YA	U-YX
105	69		U-YB	U-YY
106	6A		U-YC	U-YZ
107	6B		U-YD	U-ZA
108	6C		U-YE	U-ZB
109	6D		U-YF	U-ZC
110	6E		U-YG	U-ZD
111	6F		U-YH	U-ZE
112	70		U-YI	U-ZF
113	71		U-YJ	U-ZG
114	72		U-YK	U-ZH
115	73		U-YL	U-ZI
116	74		U-YM	U-ZJ
117	75		U-YN	U-ZK
118	76		U-YO	U-ZL

Decimal	Hex	Current Use	Proposed (Basic Option)	Proposed (Full Option)
119	77		U-YP	U-ZM
120	78		U-YQ	U-ZN
121	79		U-YR	U-ZO
122	7A		U-YS	U-ZP
123	7B		U-YT	U-ZQ
124	7C		U-YU	U-ZR
125	7D		U-YV	U-ZS
126	7E		U-YW	U-ZT
127	7F		U-YX	U-ZU

Compatibility Considerations

Implementing these changes does not change data structure definitions, message formats, or change the way the Kronos works from the front panel. Only the “Bank/Program Select” dialog (which is used many places) would be modified in the User Interface. The only backward compatibility issue would be with PCG files or MIDI messages referencing the extended banks being loaded into an un-extended Kronos. The version was incremented in the change from OS V2.x to OS 3, and newer PCG files cannot be loaded into older OS Kronos. The same revision increment could be used for this enhancement to prevent issues.

RAM considerations:

Can the Kronos support these additional banks? Given the relatively long amount of time taken to write programs and combis, it is likely that the Kronos uses Flash memory for the program and combi storage. There is no way to know how much is available, but given that modern Flash memory chips come in increments of GiBytes, it is hard to imagine that 44 additional MiBytes could not be found.

Table 3 - Memory Requirements

3706	Bytes per program		
128	Programs per bank		
463.25	KiBytes per Bank		
31	Banks today (including non-user bank G)		
14.02417	MiBytes for todays Program Banks		
81	Banks proposed with Basic Option		
36.6438	MiBytes required for Basic Option Program Banks	22.6	additional
128	Banks proposed with Full Option		
57.90625	MiBytes required for Full Option Program Banks	43.9	additional