Controlling the Korg X5DR Synthesizer from a UNIX Program

Steven Robbins

September, 1995

Technical Report CS-95-12

Abstract. This is the second in a series of technical reports dealing with the use of sound by programs. The first report dealt with the basics of MIDI, the Musical Instrument Digital Interface. In this report we describe a C library for controlling a MIDI device from a program. Although most of the library should work with any MIDI device, we deal specifically with the Korg X5DR. This device is particularly simple to interface to a computer because it has a serial input port.

Division of Computer Science
The University of Texas at San Antonio
San Antonio, TX 78249

1 Introduction

This technical report describes how to use the Korg X5DR synthesizer to produce musical sounds from a C program running on a UNIX system.

The Korg X5DR is a standard MIDI synthesizer which is among a new breed of MIDI devices that can be controlled directly through a serial port. This port makes it easy to interface to any computer having a serial port. The exact equipment needed is:

Korg X5DR ($550)
Korg Computer Interface Cable, part number AG-001 ($20)
Headphones or amplified speakers ($5 and up)

The reader is assumed to be familiar with MIDI devices. If not, see [1] for a brief introduction to MIDI for computer scientists.

Before the X5DR can be used, it must be turned on and set for general MIDI by pushing the `GLOBAL MULTI` button on the front panel. Remember to push this button after the device has initialized itself each time you turn it on.

Next the computer serial port must be set to the baud rate appropriate to the X5DR, which is 38400. Under Linux Slackware version 2.2 the first serial port (COM1) is `/dev/ttyS0` and under Solaris 2, use `/dev/term/a` for the first serial port and `/dev/term/b` for the second one.

Next the channels must be initialized. You can use up to 16 channels simultaneously with the X5DR, and each channel can represent a different instrument. We refer to the channels with numbers 0 through 15, and these can be reassigned to different instruments at any time. Initially, when the device is turned on, all channels except one are set to Program G01 which is the standard general MIDI piano. Channel 9 is set to Drum Program G129 which can be used to create a number of drum and bell sounds as well as a few sound effects.

2 Numbering Schemes

The Korg documentation refers to channel numbers 1 through 16, but when communicating with a MIDI device, the channel is encoded as a number between 0 and 15. We will assume that channels are numbered starting at zero, and all interfaces will use this numbering. Instruments, or *programs*, as they are referred to in the MIDI terminology are often referenced by a letter and a number. For example G01 represents the first General MIDI instrument, a piano. However, when a command is sent to a MIDI device associating a program with a channel, the numbers start at 0. We will use this numbering scheme so that the first MIDI program will be instrument 0. With this numbering scheme channel 9 of the Korg X5DR is set to Drum Program G129 by default.

3 rtmidlib

This section documents the `rtmidilib` program. The `rtmidilib.c` file contains all of the routines needed to initialize the Korg X5DR and generate sounds.

```
int init_port(char *portname)
```

This function initializes the port and sets the baud rate using the POSIX functions `tcgetattr`, `cfsetospeed`, and `tcsetattr`. It has been tested both under Linux which is based on BSD and Solaris 2 which is based on SVR4. Under Linux, use `/dev/ttyS0` for the first serial port (COM1) and `/dev/ttyS1` for the second one. Under Solaris 2 use `/dev/term/a` for the first serial port and `/dev/term/b` for the second one.

```
void init_channel(int chan, int instr)
```

This function sets the MIDI device so that the given channel corresponds to the given instrument. Valid channel numbers are 1 through 16 and valid instruments start at 0. A list of the instruments is given in Section 6. No error checking is done to make sure the parameters are in the proper range, but no action is taken if the channel is negative.

```
void start_note(int pitch, int velocity, int channel)
```

This function starts playing the note with the given pitch on the given channel with the given velocity. The value of `pitch` can be between 0 and 127. The velocity roughly corresponds to the volume of the note and can be between 0 (no sound) and 127 (loudest). The note will continue to play until it is turned off. For some instruments, like the violin, the sound of the note continues at almost constant volume until the note is turned off. For most instructions, including the piano, the sound dies out quickly, but it should be turned off in any case. No action is taken if the channel is negative.

```
void end_note(int pitch, int velocity, int channel)
```

This function stops the given note. It is assumed that the note had been previously started. No action is taken if the channel is negative.

```
void ms_sleep(int ms)
```

This function is used to sleep for a certain number of milliseconds. It is implemented with `select` so it should be fairly portable.

Prototypes for the above functions can be found in `rtmidilib.h`.

The following is a complete listing of `rtmidilib.c`.

```
/* rtmidilib.c */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <termio.h>
#include <sys/time.h>
#include <sys/types.h>

static int sfd;

/* returns 0 if OK or -1 if error */
int init_port(char *portname)
{
   struct termios mytermio;
   int retval;
   speed_t speed;

   sfd = open(portname,O_RDWR);
   if (sfd < 0) {
      fprintf(stderr,"Error opening sound device: %s\n",portname);
      return -1;
   }
   retval = tcgetattr(sfd,&mytermio);
   if (retval < 0) {
      fprintf(stderr,"Error getting termio structure for fd=%d\n",sfd);
      return  -1;
   }
   speed = cfgetospeed(&mytermio);
/*   fprintf(stderr,"Old speed paramter is %d\n",(int)speed);*/
   retval = cfsetospeed(&mytermio,B38400);
   if (retval < 0) {
      fprintf(stderr,"Error setting baud rate in termio structure\n");
      return -1;
   }
   retval = tcsetattr(sfd,TCSANOW,&mytermio);
   if (retval < 0) {
      fprintf(stderr,
         "Error setting new termio structure of sound output\n");
      return -1;
   }
```

```
   retval = tcgetattr(sfd,&mytermio);
   if (retval < 0) {
      fprintf(stderr,
         "Error getting termio structure for fd=%d second time\n",
         sfd);
      return -1;
   }
   speed = cfgetospeed(&mytermio);
   return sfd;
}

static void send_three_bytes(int b1, int b2, int b3)
{
   unsigned char msg[3];
   msg[0]=b1;
   msg[1]=b2;
   msg[2]=b3;
   write(sfd,msg,3);
}

/* Use select to sleep ms milliseconds */
void ms_sleep(int ms)
{
   struct timeval timeout;
   timeout.tv_sec = ms/1000;
   timeout.tv_usec  = 1000*(ms - 1000*timeout.tv_sec);
   select(10,(fd_set *)NULL,(fd_set *)NULL,(fd_set *)NULL,&timeout);
}

void init_channel(int chan, int instr)
{
   if (chan < 0) return;
   send_three_bytes(0xc0+chan,0,instr);
/*   fprintf(stderr,"channel %d set for instrument %d\n",chan,instr);*/
}

void start_note(int pitch, int velocity, int channel)
{
   if (channel < 0) return;
   send_three_bytes(0x90+channel,pitch,velocity);
}
```

```
void end_note(int pitch, int velocity, int channel)
{
   if (channel < 0) return;
   send_three_bytes(0x80+channel,pitch,velocity);
}
```

4 rttest.c

The program `rttest.c` is a simple main program which can be linked to `rtmidilib` to illustrate how the library can be used. The program takes one required command line argument giving the pitch of a note to be produced. Up to three additional command line arguments can specify the instrument, volume, and duration of the note. The default instrument is the piano, the default volume is 100 (out of 127) and the default duration is 1 second. The duration parameter is given in milliseconds.

```
/* rttest.c */

#include <stdio.h>
#include <stdlib.h>
#include "rtmidilib.h"

#define MIDIPORT "/dev/ttyS0"

#include "instrs_def.h"

char *note_to_string(int note)
{
   static char note_str[15];
   if (note < 0) return("Unknown");
   if (note > 127) return("Unknown");
   sprintf(note_str,"%s %d",note_list[note%12],note/12 - 2);
   return(note_str);
}

void main(int argc, char *argv[])
{
   int pitch;
   int instrument;
   int volume;
   int duration;
   int chan = 2;
```

```
    if (argc < 2) {
      fprintf(stderr,"Usage: %s pitch [instrument [volume [duration]]]\n",
         argv[0]);
      exit(1);
    }
    if (init_port(MIDIPORT) < 0) {
       fprintf(stderr,"Cannot initialize MIDI port %s\n",MIDIPORT);
       exit(1);
    }
    pitch = atoi(argv[1]);
    instrument = 0;
    volume = 100;
    duration = 1000;

    if (argc > 2) instrument = atoi(argv[2]);
    if (argc > 3) volume = atoi(argv[3]);
    if (argc > 4) duration =  (int) 1000*atof(argv[4]);

    if (instrument < 0) {
      fprintf(stderr,"Instrument = %d is too small\n",instrument + 1);
      exit(1);
    }
    if (instrument == 128) {
       fprintf(stderr,
           "Outputing note: p=%d(%s), instr=%d(%s), vol=%d, dur=%d\n",
           pitch,drums[pitch],instrument,instrs[instrument],
           volume,duration);
        chan = 9;
    }
    else {
       fprintf(stderr,
           "Outputing note: p=%d(%s), instr=%d(%s), vol=%d, dur=%d\n",
           pitch,note_to_string(pitch),instrument,instrs[instrument],
           volume,duration);
       init_channel(chan,instrument);
    }
    start_note(pitch,volume,chan);
    ms_sleep(duration);
    end_note(pitch,volume,chan);
}
```

5 Instrument Names

The file `instrs_def.h` defines an array of instrument names which can be used to convert instrument numbers into a recognizable form. There is also an array to convert drum kit notes into a drum name. Lastly, there is an array that can be used for displaying the names of notes. The use of these is illustrated in the `rttest` program.

```
char *instrs[129]={
"Piano",           "Brite Piano",     "Hammer Piano",    "Honky Tonk",
"New Timesh",      "Digi Piano",      "Harpssichord",    "Clavichord",
"Celesta",         "Glocken",         "Music Box",       "Vibes",
"Marimba",         "Xylophone",       "Tubular",         "Santur",
"Full Organ",      "Perc Organ",      "BX - 3 Organ",    "Church Pipe",
"Positive",        "Musette",         "Harmonica",       "Tango",
"Classic Guitar",  "Acoustic Guitar", "Jazz Guitar",     "Clean Guitar",
"Mute Guitar",     "over Drive",      "Dist Guitar",     "Rock Monics",
"Jass Bass",       "Deep Bass",       "Pick Bass",       "Tretless",
"Slap Bass 1",     "Slap Bass 2",     "Synth Bass 1",    "Synth Bass 2",
"Violin",          "Viola",           "Cello",           "Contra Bass",
"Tremolo Strings", "Pizzicato",       "Harp",            "Timpani",
"Marcato",         "Slow String",     "annalog Pad",     "String Pad",
"choir",           "Doo Voice",       "Voices",          "Orch Hot",
"Trumpet",         "Trombone",        "Tuba",            "Muted Trumpet",
"French Horn",     "Brass",           "Syn Brass 1",     "Syn Brass 2",
"Soprano Sax",     "Alto Sax",        "Tenor Sax",       "Baritone Sax",
"Sweet Oboe",      "English Horn",    "Bassoon Oboe",    "Clarinet",
"Piccolo",         "Flute",           "Recorder",        "Pan Flute",
"Bottle",          "Shakuhachi",      "Whistle",         "Ocarina",
"Square Wave",     "Saw Wave",        "Syn Caliope",     "Syn Chiff",
"Charang",         "Air Chorus",      "Rezzo 4ths",      "Bass & Lead",
"Fantaasio",       "Warm Pad",        "Poly Pad",        "Hhost Pad",
"Bowed Glass",     "Metal Pad",       "Halo Pad",        "Sweep",
"Ice Rain",        "Sound Track",     "Crystal",         "Atmosphere",
"Brightness",      "Goblin",          "Echo Drop",       "Star Theme",
"Sitar",           "Banjo",           "Shaaamisen",      "Koto",
"Kalimba",         "Scotland",        "Fiddle",          "Sshanai",
"Metal Bell",      "Agogo",           "Steel Drums",     "Wood Block",
"Taiko",           "Tom",             "Synth Tom",       "Rev Cymbol",
"Fret Noise",      "Noise Cliff",     "Seashore",        "Birds",
"Telephone",       "Helicopter",      "Stadium",         "Gun Shot",
"Drum Kit"
};
```

```
char *drums[128]={
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Rock Kick",  "Snare 3",    "Open HH",    "Fat Kick",
"Timbales",   "Snare 1",    "RollSnare1", "Real Kick",
"ProcesKick", "Side Kick",  "Rock Snare", "Hand Claps",
"LightSnare",  "Tom Lo",    "Tite HH",    "Tom Lo",
"Pedal HH",   "Tom Lo",     "Open HH",    "Tom Hi",
"Tom Hi",     "Crash Cym",  "Tom Hi",     "Ride Edge",
"China Cym",  "Ride Cup",   "Tambourine", "Splash Cym",
"Cowbell",    "Crash Cym",  "Vibraslap",  "Ride Cym 1",
"Hi Bongo",   "Lo Bongo",   "Mute Conga", "Open Conga",
"Open Conga", "Hi Timbal",  "Lo Timbal",  "Agogo",
"Agogo",      "Cabasa",     "Maracas",    "Whistle S",
"Whistle L",  "Guiro S",    "Guiro L",    "Claves",
"WoodBlock2", "WoodBlock3", "Mute Cuica", "Open Cuica",
"MuteTriang", "OpenTriang", "Cabasa",     "JingleBell",
"Bell Tree",  "Castanet",   "Side Kick",  "Taiko Lo",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown",
"Unknown",    "Unknown",    "Unknown",    "Unknown"
};

char *note_list[12]={
   "C","C#","D","D#","E","F","F#","G","G#","A","A#","B"};
```

6 Tables

The table given on the next page lists the correspondence between numeric values and the things they can represent. For each number in the range 0 to 127, 5 values are given. The first is the instrument that is represented by the number when the number is used in a channel initialization command. Next is my perception of how the volume of the note behaves while the note is on. This is called the envelope of the note and is in the column labeled L. I have used three classifications. F means that the note dies out fast. S means that the note dies out slowly. C means that the volume is constant as long as the note is on so that it does not die out until it is stopped. The third entry is the Drum Kit sound that the number corresponds to if it is used as the pitch for a note on channel 9. Only notes 28-87 represent valid Drum Kit sounds. Next is my perception of the envelope of that Drum Kit sound. Last is the note that the number represents when used as the pitch of a start note or stop note command.

I have picked out a few of the instruments and Drum Kit sounds as being an interesting subset to examine when determining what to use for an auralization. These are listed in boldface. The serious user will want to listen to all of the MIDI instruments and drum sounds.

References

[1] S. Robbins, “A MIDI Primer for Computer Scientists,” UTSA Division of Computer Science Technical Report, CS-95-11.

Num	Instrument	L	Drum	L	Note	Num	Instrument	L	Drum	L	Note
0	**G01 Piano**	**M**			C -2	64	G65 Soprano Sax	C	Open Conga	F	E 3
1	G02 Brite Piano	M			C# -2	65	G66 Alto Sax	C	Hi Timbal	F	F 3
2	G03 Hammer Piano	M			D -2	66	G67 Tenor Sax	C	Lo Timbal	F	F# 3
3	G04 Honky Tonk	M			D# -2	67	G68 Baritone Sax	C	Agogo	F	G 3
4	G05 New Timesh	M			E -2	68	**G69 Sweet Oboe**	**C**	Agogo	F	G# 3
5	G06 Digi Piano	M			F -2	69	G70 English Horn	C	**Cabasa**	**F**	A 3
6	G07 Harpssichord	M			F# -2	70	G71 Bassoon Oboe	C	Maracas	F	A# 3
7	G08 Clavichord	M			G -2	71	**G72 Clarinet**	**C**	**Whistle S**	**F**	B 3
8	**G09 Celesta**	**M**			G# -2	72	**G73 Piccolo**	**C**	**Whistle L**	**M**	C 4
9	G10 Glocken	M			A -2	73	G74 Flute	C	**Guiro S**	**F**	C# 4
10	G11 Music Box	M			A# -2	74	G75 Recorder	C	**Guiro L**	**M**	D 4
11	**G12 Vibes**	**M**			B -2	75	G76 Pan Flute	C	Claves	F	D# 4
12	**G13 Marimba**	**M**			C -1	76	G77 Bottle	C	**WoodBlock2**	**F**	E 4
13	G14 Xylophone	M			C# -1	77	G78 Shakuhachi	C	WoodBlock3	F	F 4
14	G15 Tubular	M			D -1	78	**G79 Whistle**	**C**	**Mute Cuica**	**F**	F# 4
15	G16 Santur	M			D# -1	79	G80 Ocarina	C	**Open Cuica**	**F**	G 4
16	**G17 Full Organ**	**C**			E -1	80	G81 Square Wave	C	MuteTriang	F	G# 4
17	G18 Perc Organ	C			F -1	81	G82 Saw Wave	C	**OpenTriang**	**F**	A 4
18	G19 BX - 3 Organ	C			F# -1	82	G83 Syn Caliope	C	Cabasa	F	A# 4
19	**G20 Church Pipe**	**C**			G -1	83	G84 Syn Chiff	C	JingleBell	M	B 4
20	G21 Positive	C			G# -1	84	G85 Charang	M	Bell Tree	F	C 5
21	G22 Musette	C			A -1	85	G86 Air Chorus	C	Castanet	F	C# 5
22	**G23 Harmonica**	**C**			A# -1	86	**G87 Rezzo 4ths**	**C**	Side Kick	F	D 5
23	G24 Tango	C			B -1	87	G88 Bass & Lead	C	Taiko Lo	F	D# 5
24	**G25 Classic Guitar**	**M**			C -0	88	G89 Fantaasio	C			E 5
25	G26 Acoustic Guitar	M			C# -0	89	G90 Warm Pad	C			F 5
26	G27 Jazz Guitar	M			D -0	90	G91 Poly Pad	C			F# 5
27	G28 Clean Guitar	M			D# -0	91	G92 Hhost Pad	C			G 5
28	G29 Mute Guitar	M	**Rock Kick**	**F**	E -0	92	**G93 Bowed Glass**	**C**			G# 5
29	G30 Over Drive	M	Snare 3	F	F -0	93	G94 Metal Pad	C			A 5
30	G31 Dist Guitar	M	Open HH	F	F# -0	94	G95 Halo Pad	C			A# 5
31	**G32 Rock Monics**	**C**	Fat Kick	F	G -0	95	G96 Sweep	C			B 5
32	G33 Jass Bass	M	Timbales	F	G# 0	96	**G97 Ice Rain**	**M**			C 6
33	G34 Deep Bass	M	Snare 1	F	A 0	97	G98 Sound Track	C			C# 6
34	G35 Pick Bass	M	**RollSnare1**	**M**	A# 0	98	G99 Crystal	M			D 6
35	G36 Fretless	M	Real Kick	F	B 0	99	G100 Atmosphere	C			D# 6
36	G37 Slap Bass 1	M	ProcesKick	F	C 1	100	G101 Brightness	C			E 6
37	G38 Slap Bass 2	M	Side Kick	F	C# 1	101	G102 Goblin	C			F 6
38	G39 Synth Bass 1	M	**Rock Snare**	**F**	D 1	102	G103 Echo Drop	C			F# 6
39	G40 Synth Bass 2	C	**Hand Claps**	**F**	D# 1	103	G104 Star Theme	C			G 6
40	**G41 Violin**	**C**	LightSnare	F	E 1	104	G105 Sitar	M			G# 6
41	G42 Viola	C	Tom Lo	F	F 1	105	G106 Banjo	M			A 6
42	G43 Cello	C	**Tite HH**	**F**	F# 1	106	G107 Shamisen	M			A# 6
43	G44 Contra Bass	C	Tom Lo	F	G 1	107	G108 Koto	M			B 6
44	**G45 Tremolo Strings**	**C**	Pedal HH	F	G# 1	108	**G109 Kalimba**	**M**			C 7
45	**G46 Pizzicato**	**F**	Tom Lo	F	A 1	109	G110 Scotland	C			C# 7
46	G47 Harp	M	Open HH	**F**	A# 1	110	G111 Fiddle	C			D 7
47	**G48 Timpani**	C	Tom Hi	F	B 1	111	G112 Shanai	C			D# 7
48	**G49 Marcato**	**C**	Tom Hi	F	C 2	112	G113 Metal Bell	M			E 7
49	G50 Slow String	C	**Crash Cym**	**M**	C# 2	113	**G114 Agogo**	**M**			F 7
50	**G51 Annalog Pad**	C	**Tom Hi**	F	D 2	114	G115 Steel Drums	M			F# 7
51	G52 String Pad	C	Ride Edge	F	D# 2	115	**G116 Wood Block**	**M**			G 7
52	G53 Choir	C	**China Cym**	**M**	E 2	116	**G117 Taiko**	**M**			G# 7
53	G54 Doo Voice	C	**Ride Cup**	**F**	F 2	117	G118 Tom	M			A 7
54	G55 Voices	C	Tambourine	F	F# 2	118	G119 Synth Tom	M			A# 7
55	**G56 Orch Hot**	**F**	**Splash Cym**	M	**G 2**	119	**G120 Rev Cymbol**	**M**			B 7
56	**G57 Trumpet**	C	**Cowbell**	**F**	G# 2	120	**G121 Fret Noise**	**M**			C 8
57	G58 Trombone	C	Crash Cym	F	A 2	121	G122 Noise Cliff	M			C# 8
58	G59 Tuba	C	**Vibraslap**	**M**	A# 2	122	**G123 Seashore**	**M**			D 8
59	G60 Muted Trumpet	C	Ride Cym 1	M	B 2	123	**G124 Birds**	**C**			D# 8
60	G61 French Horn	C	Hi Bongo	F	C 3	124	**G125 Telephone**	**C**			E 8
61	G62 Brass	C	Lo Bongo	F	C# 3	125	**G126 Helicopter**	**C**			F 8
62	G63 Syn Brass 1	C	Mute Conga	F	D 3	126	**G127 Stadium**	**C**			F# 8
63	G64 Syn Brass 2	C	Open Conga	F	D# 3	127	**G128 Gun Shot**	**M**			G 8