Programmer’s Guide to Yamaha YM2154 (RYP4) Rhythm Processor

Version 1.1 Jun 5, 2021

Written by Jari Kangas

Summary

This document contains a programmer’s guide and a minimal data sheet for Yamaha RYP4 Rhythm Processor chip YM2154. Also its supporting chips YM3012/YM3010 and YM2190x are shortly described.

Disclaimer

Because most of the information presented here is based on reverse engineering, it is very probable that there are misunderstandings and omissions. The writer assumes no responsibility for any damages arising out of use of this text. No warranty is provided about correctness of any information in this file.

Table of Contents

1 Introduction

This document contains a programmer's guide and a minimal data sheet for Yamaha YM2154, which Yamaha calls RYP4 (Rhythm Processor 4). Also supporting chips YM3012/3010 (DACs) and YM2190x (sample ROMs) are shortly described.

There is no official information available for these chips, except for YM3012. Everything presented in this document is based on these sources:

- Reverse engineering of Yamaha PSR-70 keyboard, by the author
  - https://retroandreverse.blogspot.com/search/label/PSR-70%20reverse%20engineering
- Yamaha PSR-70 Owner's Guide, by Yamaha Corporation 1985
  - http://www.synthmanuals.com/manuals/yamaha/psr-70/owners_manual/
- Yamaha PSR-70 Service Manual, by Yamaha Corporation 1985
  - https://elektrotanya.com/yamaha_psr-70_sm.pdf/download.html
- Yamaha RX-11 Service Manual, by Yamaha Corporation 1984
  - https://elektrotanya.com/yamaha_rx11_sm.pdf/download.html

One of the main information sources is the PSR-70 original internal software, which I have studied closely. I will be referring to it several times with the term "PSR-70 firmware".

As a comparison and reference, I have used the Yamaha RX-11 drum machine schematics from the service manual. Besides the schematic, it contains many useful pieces on information.

The RYP4 chip is used at least in PSR-60/70/80 keyboards and RX-11/15/21 drum machines.

2 The RYP4 drum chip

The RYP4 is a multi-function chip: besides the actual sample player, it contains a 10-channel A/D-converter, four general purpose output pins and a timer capable of generating interrupts.

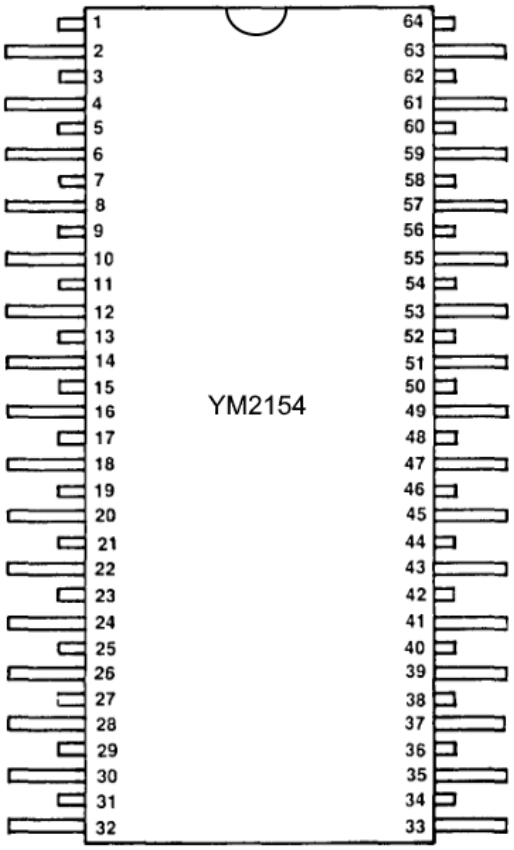
RYP4 chip contains no D/A-converter. It outputs serial or parallel digital data to an external DAC. In PSR-70, an YM3012 is used for this purpose. An official Yamaha datasheet for YM3012 is available in internet.

The data for drum samples are contained in separate ROM chips. It is possible to connect several ROMs to the RYP4.

The RYP4 is meant to be connected to a normal processor bus. Data bus is 8-bit wide and there are 7 address lines. Bus control signals are the normal $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$. The chip can also request an interrupt.

There is no bus timing information available. In PSR-70, a Z80 CPU with 6 MHz clock is used and the RYP4 is mapped as a part of I/O address space. There is a wait-state logic implemented in hardware, and it seems to add 3 clocks wait when accessing the RYP4. This gives some rough idea about the bus timing specifications.

The IC comes in a 64-pin quad-in-line (QIL) package. Pin numbering is according to this picture:



Pin functions are as follows (the table is from RX-11 service manual).

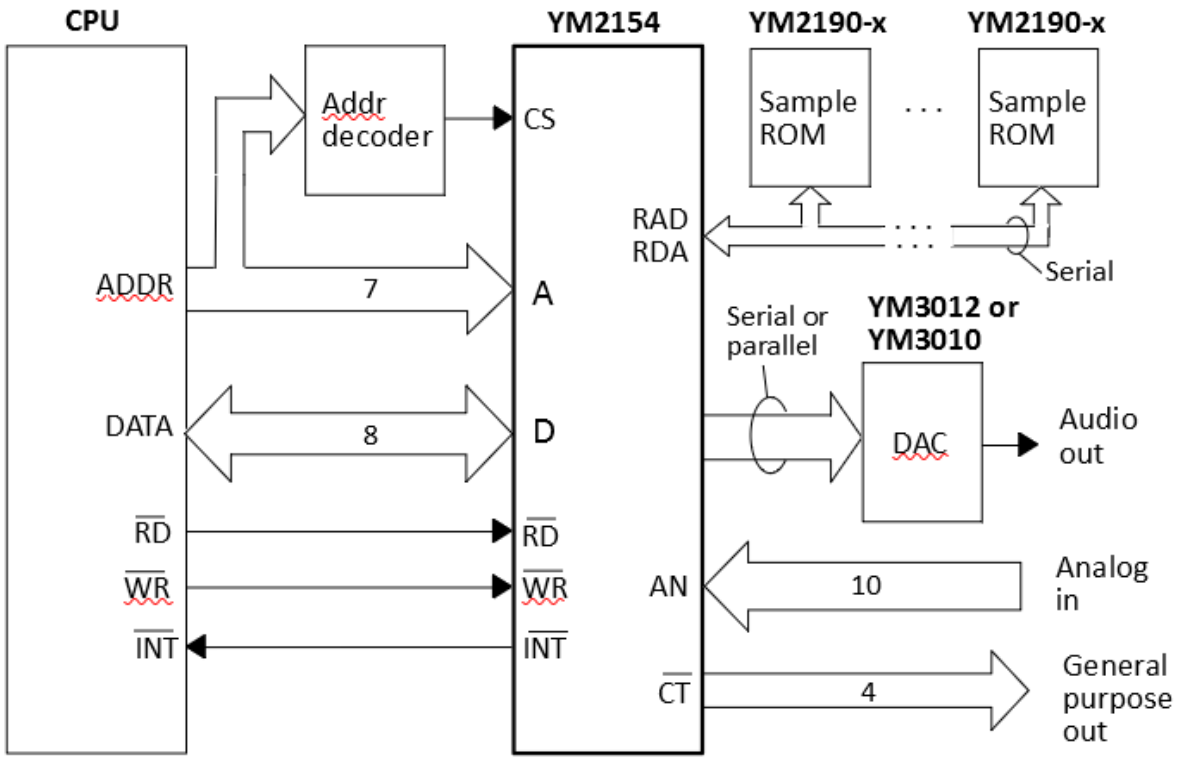
YM2154 RYP-4

PIN NO.	NAME	I/O	FUNCTION	PIN NO.	NAME	I/O	FUNCTION
1	Vss	I	Digital ground	34	$\overline{CT_1}$/ SH_3	O	CT_1/LED control / SH_3/Sample and hold
2	$\overline{INT}$	O	Interrupt request				
3	S_2	O	Exponent Output to DAC	35	SHR/ SH_2	O	SHR/Sample and hold / SH_2/ "
4	S_1	O					
5	S_0	O		36	SHL/ SH_1	O	SHL/ " / SH_1/ "
6	B_9	O	Mantissa Output to DAC	37	$\overline{CTO}$	O	LED control
7	B_8	O		38	$\overline{RD}$	I	Read Enable
8	B_7	O		39	$\overline{WR}$	I	Write Enable
9	B_6	O		40	$\overline{CS}$	I	Chip select
10	B_5	O		41	A_0	I	Address bus
11	B_4	O		42	A_1	I	
12	B_3	O		43	A_2	I	
13	B_2	O		44	A_3	I	
14	B_1	O		45	A_4	I	
15	ϕB/B0	O	ϕB/Clock for DAC B0/Mantissa (LSB)	46	A_5	I	
16	SD	O	Serial data output	47	A_6	I	
17	Vss	I	Digital ground	48	D_0	I/O	Data bus
18	VREF	I	Reference voltage for ADC	49	D_1	I/O	
19	VDD2	I	Analog DC supply	50	D_2	I/O	
20	VGND	I	Analog ground	51	D_3	I/O	
21	AN_1	I	Analog data in	52	D_4	I/O	
22	AN_2	I		53	D_5	I/O	
23	AN_3	I		54	D_6	I/O	
24	AN_4	I		55	D_7	I/O	
25	AN_5	I		56	RDA2	I	ROM data 2 (ch7 ~ 12)
26	AN_6	I		57	RDA1	I	" 1 (ch1 ~ 6)
27	AN_7	I		58	RSYNC	O	ROM data synchro pulse
28	AN_8	I		59	RAD2	O	ROM address 2 (ch7 ~ 12)
29	AN_9	I		60	RAD1	O	" 1 (ch1 ~ 6)
30	AN_{10}	I		61	VDD	I	Digital DC supply (+5V)
31	$\overline{IC}$	I	Initial clear	62	ϕ_1 OUT	O	ROM CLOCK
32	$\overline{CT_3}$/ ST	O	CT_3/LED control ST/Strobe DAC data	63	ϕM_2	I	Master clock pulse
33	$\overline{CT_2}$/ SH_4	O	CT_2/ " SH_4/Sample and hold	64	ϕM_1	O	"

Table 1. Pin functions

3 RYP4 functionality

A basic RYP4 system is depicted in the picture below. Besides the controlling CPU, an external D/A-converter and at least one sample ROM are needed.



3.1 Sample player and sample ROMs

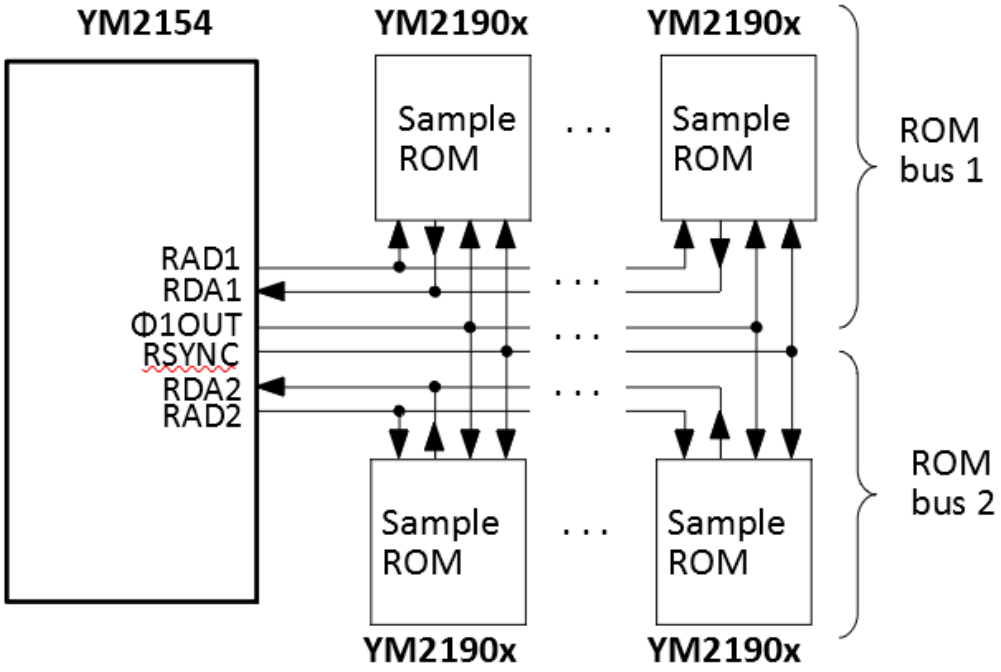
RYP4 does not contain any samples, it is only a player for externally stored samples. The player has 12 channels, meaning it can play maximum of 12 samples simultaneously. The channels are divided into two groups, group A containing channels 1 to 6, group B channels 7 to 12.

3.1.1 Sample ROM connections and timing

Sample ROMs are Yamaha's own special ROMs, no generic ROMs can be used. They connect to RYP4 using serial bus, which has 4 signals:

- RAD = ROM address, from RYP4 to ROM
- RDA = ROM data, from ROM to RYP4
- RSYNC = byte sync, from RYP4 to ROM
- Φ1 OUT = bus clock, from RYP4 to ROM

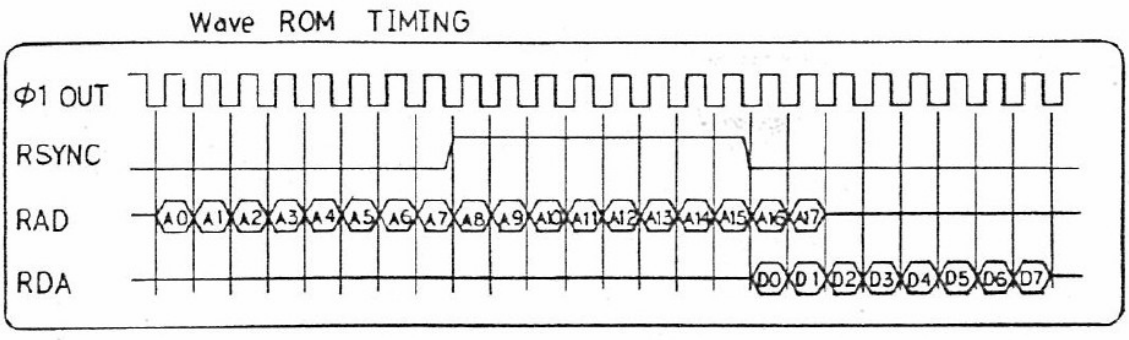
There are two separate ROM buses. The RAD and RDA signals are bus specific, RSYNC and Φ1 OUT are common. ROMs are connected a follows:



Sample player channels 1...6 play samples only from ROMs connected to bus 1, channels 7...12 play samples only from ROMs in bus 2.

Each bus can have several ROMs connected. I don't know what is the maximum, but in RX-11 there are four ROMs in bus 1 and two in bus 2. In PSR-70, both buses have one ROM.

This is the bus timing diagram from RX-11 service manual:



The bus clock frequency is the same as the crystal frequency in pins ΦM1 and ΦM2. In PSR-70 the crystal is 2.25 MHz and in RX-11 it is 2.7 MHz.

The timing picture does not fully match with my measurements. Actually, one RSYNC cycle is 18 cycles of Φ1 OUT. This gives RSYNC frequency of 125 kHz for PSR-70, which matches to measurements.

One full RSYNC cycle is one fetch from memory. Because there may be six channels playing at the same time, each channel gets a new sample at a rate of 20.83 kHz (125/6). So this is the basic sampling frequency.

The same calculation leads to sampling frequency of 25 kHz in RX-11.

3.1.2 Sample ROM contents

There exists at least 8 different sample ROMs for RYP4, designated as YM2190x, where x is 1...3, 5...9. RX-11 contains six of these, RX-15 four. Each ROM contains 32 KB of sample data.

This table is from RX-11 service manual:

WAVE ROM SOUND SOURCE

WAVE ROM NO.	SOUND SOURCE	MEMORY CAR (BYTES)	
YM 21901	RIDE	32K	same RX-15
YM 21902	CRASH	32K	
YM 21903	HH OPEN 1	20K	
	PEDAL	2K	
	CLOSED 1	3K	
	CLAPS 1	3K	
	COWBELL 1	3K	
	TEST	1K	
YM 21907	BD (M1)	2K	
	SD (M)	4K	
	SD (HT1)	4K	
	RIMSHOT 1	2K	
	SHAKER	2K	
	TOM 1	6K	
	TOM 2	6K	
	TOM 3	6K	
YM 21906	HH OPEN 2	22K	
	CLOSED 2	3K	
	CLAPS 2	3K	
	COWBELL 2	3K	
	NOT USED	1K	
YM 21905	TOM 4	6K	
	BD (M2)	3K	
	BD (HV)	2K	
	SD (L)	3K	
	SD (HV)	5K	
	SD (HT2)	3K	
	SD (HT3)	3K	
	SD (HT4)	3K	
	SD (HT5)	3K	
	RIMSHOT 2	1K	
	TEST	1K	

Table 2. ROM contents for ROMs 1...3, 5...7

PSR-70 contains none of these ROMs, it contains YM21908 and YM21909. I have not found similar table for them, so I made it myself, see table 3 below.

ROM 8 looks kind of a ”poor mans drum kit”, where the whole drum kit is squeezed to one ROM. If we compare it to RX-11 or RX-15, where a ride cymbal alone fills one ROM, it is quite a big difference. In ROM 8 the ride cymbal fills half of the ROM and the rest of the whole drum kit is stuffed in the other half. They left out the crash cymbal completely, no way it would have fitted in the same ROM. All samples are also a bit smaller compared to RX-11 ROMs.

ROM 9 contains mostly Latin percussions and one “normal” drum kit sound, the Tom 3.

It is very probable that ROMs 8 and 9 cannot be used together with other ROMs, because ROMs 8 and 9 have been sampled at a lower rate (20.83 kHz vs. 25 kHz). This also explains the smaller sample sizes, and suggests lower sound quality.

This is my reverse engineered sound table for ROMs 8 and 9. The extra columns are explained in later chapters.

ROM nr	Instrument	Sample size (kbytes)	Sample length (ms)	Sample start block (hex)	Sample end block (hex)	SS1 (hex)	SS2 (hex)	SS3 (hex)
YM21908	Ride	15.5	762	000	0F7	00	00	F7
	HH Open	6	295	0F8	157	0F	81	57
	HH Closed	0.75	37	158	163	15	81	63
	SD Hi	1	49	164	173	16	41	73
	SD Lo	1.75	86	174	18F	17	41	8F
	Rimshot	0.5	25	190	197	19	01	97
	BD	1	49	198	1A7	19	81	A7
	Tom 1	2.5	123	1A8	1CF	1A	81	CF
	Tom 2	3	147	1D0	1FF	1D	01	FF
YM21909	Tom 3	7.5	369	000	077	00	00	77
	Bongo 1	1	49	078	087	07	80	87
	Bongo 2	0.5	25	088	08F	08	80	8F
	Bongo 3	2	98	090	0AF	09	00	AF
	Claves	0.75	37	0B0	0BB	0B	00	BB
	Timbale Hi	3	147	0BC	0EB	0B	C0	EB
	Timbale Lo	3	147	0EC	11B	0E	C1	1B
	Cuica Hi	1.75	86	11C	137	11	C1	37
	Cuica Lo	5	246	138	187	13	81	87
	Agogo Hi	1.25	61	188	19B	18	81	9B
	Agogo Lo	5	246	19C	1EB	19	C1	EB
	Claps	1.25	61	1EC	1FF	1E	C1	FF

Table 3. ROM contents for ROMs 8 and 9

3.1.3 Addressing the samples

All channels in a channel group can play any sample from the ROMs connected to group's bus. In PSR-70, the YM21908 is on bus 1 and YM21909 is on bus 2.

The ROMs have no internal data structure, which would indicate where a sample starts or ends. The samples are stored contiguously and it is RYP4 programmer's responsibility to know where the desired sample starts and ends. If these are not set correctly, you may hear the end of one sample and start of the next one, or several samples played one after each other. RYP4 just blindly plays the given area of the ROM, it does not care whether it forms a full or partial sample.

The ROM contents is addressed in blocks of 64 bytes. Each sample starts and ends at a block boundary. One 32 KB ROM contains 512 blocks, numbered from 0 to 511, or 000H to 1FFH in hex.

The PSR-70 contains only one ROM on both buses, so I don't know how the addressing goes when there are more than one ROM on the bus. My guess is that the ROMs are contiguously after each other, forming effectively one big ROM. Four 32 KB ROMs would be seen as one 128 KB ROM, with block numbers from 0 to 2047. The way how the block numbers are encoded in the RYP4 chip registers (explained later in this document) allows 4095 as the maximum block number, which would correspond to eight 32 KB ROMs on one bus.

I have not had possibility to test with RX-11 to build a start and end block table for the other ROMs, but an approximate table could be built by using the published sample lengths in table 2.

3.1.4 Playing a sample

It is possible to set any start and end block addresses to any channel, so each channel can play all samples from its own group. Playing a sample with RYP4 goes basically like this:

- Select a channel to use for playing, taking into account the group limitations
- Set start and end block numbers of the desired sample to channel's registers, according to table 3
- Trigger the channel

Once triggered, the channel plays the ROM contents from start block to end block. Playing cannot be stopped, but it can be retriggered before the end block has been reached, which causes the playing to restart from the start block.

Although playing cannot be stopped, the playback rate can be switched to very fast. I did not measure the rate. This can be used as a kind of "sample abort", by playing an already started sample quickly to end. PSR-70 firmware uses it like this to make sure the previous sample has stopped before loading a new sample to the channel.

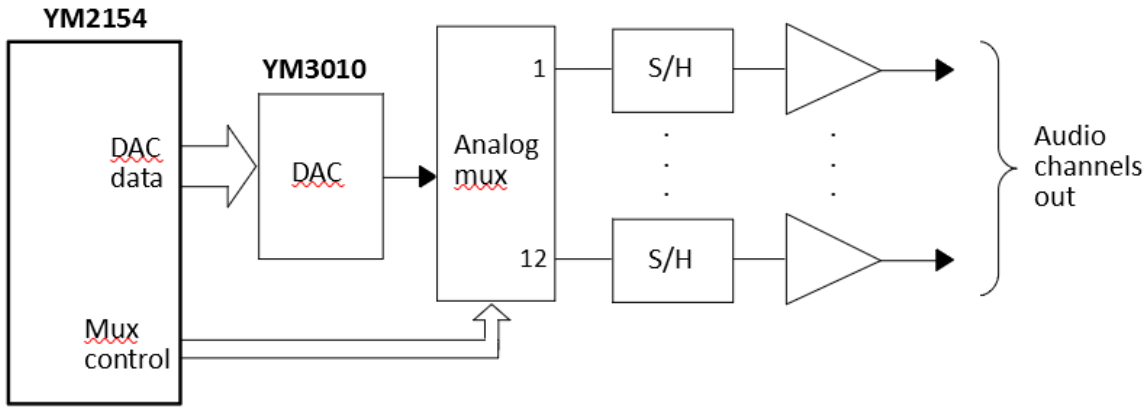
3.2 D/A converter

RYP4 chip does not contain a D/A converter, it needs an external one. There are two alternate methods for connecting a D/A converter: serial or parallel. The IC has separate output pins for both methods. Connection method can be selected by setting the register values.

In the serial method, RYP4 outputs two digital serial streams, left and right audio channels, to the DAC IC. In PSR-70, an YM3012 is used as the DAC. An official Yamaha datasheet for YM3012 is available in internet. I won't handle this more here, the datasheet describes this quite well.

The parallel method is more complex. It is used in RX-11, so I don't have any firsthand experience on this. Conclusions presented here are based on RX-11 schematic and some tests on PSR-70, setting the RYP4 to parallel DAC mode (which of course does not work, because there is no hardware for that).

The principle used in RX-11 is shown in the picture below. After the DAC there is an 12-output analog mux controlled by RYP4. In each mux output there is a sample-and-hold (S/H) circuit and a buffer amplifier. This suggests that all 12 sample player's sound channels are muxed to DAC one at a time and the DAC output is switched to a corresponding S/H circuit. This way we can get each channel to separate outputs. This also explains the parallel interface between RYP4 and DAC: the muxing has to happen so fast (twelve times during each sample period) that a serial interface would be too slow.



The main difference between the serial and parallel DAC modes:

- In serial mode, the 12 sound channels are digitally mixed inside the RYP4 and only the mixed-down result comes out as two (left and right) streams to the DAC.
- In parallel mode, it is possible (with additional hardware described above) to get all 12 sound channels separately out for external mixing.

In both modes it is possible to control the level and pan for each sound channel by writing to registers. In serial mode this is the only way for level/pan controlling, because the stereo mixing is done inside the RYP4.

Inferring from the RX-11 schematic, the digital stereo mixdown can be directed to the first two channels also in the parallel mode. I don't know how this is controlled in the RYP4 registers.

RYP4 IC pins SD, SHL and SDR are used for DAC interfacing in serial mode. In parallel mode, pins B0...B9, S0...S2 and ST control the DAC, while pins SH1...SH4 control the analog mux.

3.3 A/D converter

RYP4 contains an 8-bit A/D converter with 10 input channels in pins AN1...AN10. It is continuously converting all channels and the conversion results can be read from registers.

The A/D converter's reference value is brought to pin VREF. In PSR-70 and RX-11, a 2.5V value is used.

Both in the PSR-70 and RX-11 the A/D is used for reading the front panel potentiometers. Assuming that this is the intended usage, the A/D sample rate is probably not very high. I have not measured it.

3.4 General purpose output

There are four general purpose output pins (CT0...CT3) in the RYP4, which can be controlled through a register.

In parallel DAC mode, three of the output pins (CT1...CT3) are used for controlling the DAC and analog mux, only one pin (CT0) is available as a general output.

PSR-70 uses three of the outputs to control the chorus circuitry.

3.5 Timer and interrupts

RYP4 contains one general purpose timer which is capable of generating interrupts. Timer is the only thing inside the chip which generates interrupts.

The timer is a 11-bit auto-reload timer, clocked with approximately 21 kHz internal clock. Most propably it is using the 20.833 kHz sample clock. It counts up and when it overflows (from 7FFH to 0) it generates an interrupt and reloads the set value. The maximum interrupt interval (achieved with reload value 0) is about 98 ms.

Timer can be started and stopped and its reload value can be set by writing to registers. Interrupt is acknowledged by reading a register.

In PSR-70 the timer is used in auto-accompaniment mode to produce the basic tempo for the sequencer. It produces 24 interrupts per quarter note. The reload value is changed according to tempo setting.

4 Register reference

While describing registers, we will need hex numbers. For that purpose I will use the Intel/Zilog originating style to use postfix 'H', like 0FH. If there is no 'H', the number is decimal.

RYP4 has 128 registers, 8-bits wide, or at least 128 register addresses, not all have any functionality. Most of the registers are write-only. There are a few registers which can also be read, but the read register is always a completely different thing than the write register in the same address. That's why the write and read registers are described separately.

The registers show up as 128 consecutive addresses in the controlling processor's address space. This may be memory address space or I/O address space, depending on the hardware designer's decisions. In PSR-70, registers are in I/O address space, using upper half of the Z80's 256 register I/O address space.

4.1 Register groups

4.1.1 Write registers

Register adresses	Functionality
01H...07H	General settings and controls
08H...0DH	Sound channel pan settings for groups A and B
10H...15H	Sound channel group A level settings
18H...1DH	Sound channel group B level settings
20H...35H	Sound channel group A sample selection settings
38H...4DH	Sound channel group B sample selection settings
50H...7FH	Not used

Table 4. Write registers

4.1.2 Read registers

Register adresses	Functionality
01H...0AH	A/D values for channels 1...10
0EH	Interrupt acknowledge

Table 5. Read registers

4.1.3 Sound channel / register number relation

In all write register groups except the general settings (01H...07H) the last 3 bits of the register address indicate the sound channel. The actual channel number depends on which channel group this register is controlling.

Group	000	001	010	011	100	101	110	111
A	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	-	-
B	Ch 7	Ch 8	Ch 9	Ch 10	Ch 11	Ch 12	-	-

Table 6. Channel numbers and register adress least significant bits

An example: Table 4 says that write registers in range 38H...4DH are controlling group B settings, so all addresses in that range ending with bits 010 (i.e. addresses 3AH, 42H, 4AH) are controlling sound channel 9.

4.2 Write registers 01H...07H: General settings

This register area contains general settings and other registers that are not specific to certain sound channel.

4.2.1 Register 01H: Unknown

This register contains settings affecting to all channels. The meanings of the bits are unknown. PSR-70 firmware writes 0 to this register in the initalization and never touches it after that.

D7	D6	D5	D4	D3	D2	D1	D0
-	U1	-	-	-	-	-	U2

U1 Unknown. When this bit is set to 1 and triggering a sound channel, RYP4 produces a single pulse with same length as the selected sample length.

U2 Unknown. Setting this bit to 1, all sounds disappear. Timer and A/D continue working.

Just a guess: U2 might be a "redirect stereo mix to channels 1 and 2 in parallel DAC mode" (see chapter 3.2). I have had no possibility to test that.

4.2.2 Register 02H: Timer reload value

This register sets the RYP4 timer 11-bit reload value. It is set with two separate writes: if the highest bit is 1, the rest of the bits are interpreted as the upper 7 bits of the reload value. If the highest bit is 0, they are interpreted as the lower 4 bits. Bytes can be written in any order.

D7	D6	D5	D4	D3	D2	D1	D0
1	TV10	TV9	TV8	TV7	TV6	TV5	TV4
0	-	-	-	TV3	TV2	TV1	TV0

TV10...TV0 11-bit timer reload value, 000H...7FFH

4.2.3 Register 03H: Output and timer control

With this register the timer can be started and stopped and the general purpose outputs can be set on or off.

D7	D6	D5	D4	D3	D2	D1	D0
CT3	CT2	CT1	CT0	-	TR	-	-

CTx Control the CTx output pin state. Note that all output pins are active low, meaning that when setting CTx bit to 1, the corresponding output pin will go to low state.

TR Timer run. Setting this bit to 1 will start the timer, 0 will stop it. When timer is running, it will always generate interrupts, there is no separate interrupt enable.

4.2.4 Register 04H: Master output level

This register controls the output level of the whole RYP4. It affects to all sound channels.

D7	D6	D5	D4	D3	D2	D1	D0
-	-	ML5	ML4	ML3	ML2	ML1	ML0

ML5...ML0 Master output level 0...63. 0 = no output, 63 = max level.

4.2.5 Register 05H, 06H: Channel triggers

These two registers act as the sound channel triggers.

05H: Channel group B triggers

D7	D6	D5	D4	D3	D2	D1	D0
-	-	T12	T11	T10	T9	T8	T7

06H: Channel group A triggers

D7	D6	D5	D4	D3	D2	D1	D0
-	-	T6	T5	T4	T3	T2	T1

Tx — Trigger for sound channel x. Writing 1 to this bit triggers the corresponding sound channel's sample play. There is no need to write it back to 0, writing 0 just means "don't trigger this channel".

4.2.6 Register 07H: DAC mode

This register controls the DAC mode. After reset the DAC mode is parallel. PSR-70 firmware writes 01H to this register in the initalization and never touches it after that.

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	-	DM

DM — DAC mode. 0 = parallel, 1 = serial.

4.3 Write registers 08H...0DH: Channel pan

This register area contains sound channel pan controls. Opposite to OPQ, the panning is "real": there are 15 steps available between left and right.

The channel and register numbering deviates slightly from the principle described in chapter 4.1.3: each register controls two channels, one from each group. For example, register 0AH controls pan for channels 3 and 9.

D7	D6	D5	D4	D3	D2	D1	D0
PB3	PB2	PB1	PB0	PA3	PA2	PA1	PA0

PB3...PB0 Pan value for group B channel, 0...15.

PA3...PA0 Pan value for group A channel, 0...15.

Pan values:

0 = mute channel
1 = full left
…
8 = center
…
15 = full right

4.4 Write registers 10H...15H: Channel level / playback rate, group A

This register area contains sound channel level controls. The channel and register numbering is according to chapter 4.1.3.

D7	D6	D5	D4	D3	D2	D1	D0
-	PR2	PR1	CL4	CL3	CL2	CL1	CL0

CL4...CL0 Channel output level, 0...31. 0 = no output, 31 = max level.

PR2, PR1 Playback rate. By setting **both** of these bits to 1, the channel playback rate changes to fast. See chapter 3.1.4.

4.5 Write registers 18H...1DH: Channel level / playback rate, group B

Same as for group A, with different register numbers.

4.6 Write registers 20H...35H: Sample selection, group A

This register area defines for channel group A which sample the channel is playing when triggered. The sample selection and sample ROM addressing are discussed in chapter 3.1.3. The sample start block and end block numbers for ROMs 8 and 9 are listed in table 3. This chapter describes how those two numbers are encoded in the registers.

The channel and register numbering is according to chapter 4.1.3. Each channel has 3 registers for sample selection, I call them SS1, SS2, SS3.

Registers 20H...25H: Sample select 1 (SS1):

D7	D6	D5	D4	D3	D2	D1	D0
SB11	SB10	SB9	SB8	SB7	SB6	SB5	SB4

Registers 28H...2DH: Sample select 2 (SS2):

D7	D6	D5	D4	D3	D2	D1	D0
SB3	SB2	SB1	SB0	EB11	EB10	EB9	EB8

Registers 30H...35H: Sample select 3 (SS3):

D7	D6	D5	D4	D3	D2	D1	D0
EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0

SB11...SB0 Sample start block number, 000H...FFFH
EB11...EB0 Sample end block number, 000H...FFFH

Example: We want to play bass drum sample on channel 2.
- Table 3 gives for BD: start block = 198H, end block = 1A7H
- Encoded to sample select bytes: SS1 = 19H, SS2 = 81H, SS3 = A7H
- Table 6 gives: channel 2 register addresses end always with bits 001
- Three writes are needed:
  SS1: 19H to register 21H
  SS2: 81H to register 29H
  SS3: A7H to register 31H

For the convenience, the SS-bytes for each sample are listed in table 3 directly.

4.7 Write registers 38H...4DH: Sample selection, group B

This register area defines for channel group B which sample the channel is playing when triggered. The encoding method for SS-bytes is same as above, only register numbers are different.

Registers 38H...3DH: Sample select 1 (SS1)

Registers 40H...45H: Sample select 2 (SS2)

Registers 48H...4DH: Sample select 3 (SS3)

Example: We want to play claps sample on channel 12.
- Table 3 gives for claps: start block = 1ECH, end block = 1FFH
- Encoded to sample select bytes: SS1 = 1EH, SS2 = C1H, SS3 = FFH
- Table 6 gives: channel 12 register addresses end always with bits 101
- Three writes are needed:
  SS1: 1EH to register 3DH
  SS2: C1H to register 45H
  SS3: FFH to register 4DH

4.8 Read registers 01H...0AH: A/D-converter values

This register area contains the analog values in analog inputs AN1...AN10, converted to a 8-bit value.

The register number is same as the corresponding AN-channel number.

D7	D6	D5	D4	D3	D2	D1	D0
AV7	AV6	AV5	AV4	AV3	AV2	AV1	AV0

AV7...AV0 Analog value, 0...255

4.9 Read register 0EH: Interrupt ack / interrupt counter

By reading this register, the interrupt is acknowledged. RYP4 will keep the INT-output active until this register is read.

The read value contains an 8-bit interrupt counter. At each interrupt it is incremented by 1. The PSR-70 firmware does not use this value for anything.

D7	D6	D5	D4	D3	D2	D1	D0
IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0

IC7...IC0 Interrupt counter, 0...255

5 Register tables

5.1 Write registers

General controls

Reg	Bit								Notes
	7	6	5	4	3	2	1	0	
00H									-
01H	-	?	-					?	Unknown
02H	0/1	Timer lo/hi							Timer value
03H	CT3	CT2	CT1	CT0	-	TR	-	-	Output and timer control
04H	-	-	Level						Output level
05H	-	-	T12	T11	T10	T9	T8	T7	Triggers, group B
06H	-	-	T6	T5	T4	T3	T2	T1	Triggers, group A
07H	-	-	-	-	-	-	-	DM	DAC mode

Pan and level for channels

Reg	Bit								Notes	
	7	6	5	4	3	2	1	0		
08H	Pan ch 7				Pan ch 1					Groups A&B
09H	Pan ch 8				Pan ch 2					
0AH	Pan ch 9				Pan ch 3					
0BH	Pan ch 10				Pan ch 4					
0CH	Pan ch 11				Pan ch 5					
0DH	Pan ch 12				Pan ch 6					
0EH									-	
0FH									-	
10H	-	PB rate		Level					Channel 1	Group A
11H	-	PB rate		Level					Channel 2	
12H	-	PB rate		Level					Channel 3	
13H	-	PB rate		Level					Channel 4	
14H	-	PB rate		Level					Channel 5	
15H	-	PB rate		Level					Channel 6	
16H									-	
17H									-	
18H	-	PB rate		Level					Channel 7	Group B
19H	-	PB rate		Level					Channel 8	
1AH	-	PB rate		Level					Channel 9	
1BH	-	PB rate		Level					Channel 10	
1CH	-	PB rate		Level					Channel 11	
1DH	-	PB rate		Level					Channel 12	
1EH									-	
1FH									-	

Sample selection settings

Reg	Bit								Notes	
	7	6	5	4	3	2	1	0		
20H	Sample select 1								Channel 1	Group A
21H	Sample select 1								Channel 2	
22H	Sample select 1								Channel 3	
23H	Sample select 1								Channel 4	
24H	Sample select 1								Channel 5	
25H	Sample select 1								Channel 6	
26H									-	
27H									-	
28H	Sample select 2								Channel 1	
29H	Sample select 2								Channel 2	
2AH	Sample select 2								Channel 3	
2BH	Sample select 2								Channel 4	
2CH	Sample select 2								Channel 5	
2DH	Sample select 2								Channel 6	
2EH									-	
2FH									-	
30H	Sample select 3								Channel 1	
31H	Sample select 3								Channel 2	
32H	Sample select 3								Channel 3	
33H	Sample select 3								Channel 4	
34H	Sample select 3								Channel 5	
35H	Sample select 3								Channel 6	
36H									-	
37H									-	
38H	Sample select 1								Channel 7	Group B
39H	Sample select 1								Channel 8	
3AH	Sample select 1								Channel 9	
3BH	Sample select 1								Channel 10	
3CH	Sample select 1								Channel 11	
3DH	Sample select 1								Channel 12	
3EH									-	
3FH									-	
40H	Sample select 2								Channel 7	
41H	Sample select 2								Channel 8	
42H	Sample select 2								Channel 9	
43H	Sample select 2								Channel 10	
44H	Sample select 2								Channel 11	
45H	Sample select 2								Channel 12	
46H									-	
47H									-	
48H	Sample select 3								Channel 7	
49H	Sample select 3								Channel 8	
4AH	Sample select 3								Channel 9	
4BH	Sample select 3								Channel 10	
4CH	Sample select 3								Channel 11	
4DH	Sample select 3								Channel 12	
4EH									-	
4FH									-	

5.2 Read registers

Reg	Bit								Notes
	7	6	5	4	3	2	1	0	
00H									-
01H	A/D value								AN1
02H	A/D value								AN2
03H	A/D value								AN3
04H	A/D value								AN4
05H	A/D value								AN5
06H	A/D value								AN6
07H	A/D value								AN7
08H	A/D value								AN8
09H	A/D value								AN9
0AH	A/D value								AN10
0BH									-
0CH									-
0DH									-
0EH	Interrupt counter								Int ack / int counter
0FH									-

Version history

V 1.0	May 27, 2021	First version
V 1.1	Jun 5, 2021	Timer is 11-bit, not 14.