The CoCo PSG
User's Manual

Congratulations on your purchase of a CoCo PSG cartridge.

The CoCo PSG is a plug-in cartridge for use with Tandy Color Computers, and can also be used with Dragon systems. The 'PSG' in CoCo PSG stands for 'Programmable Sound Generator'. This cartridge adds special sound features and more to enhance the computer's capability.

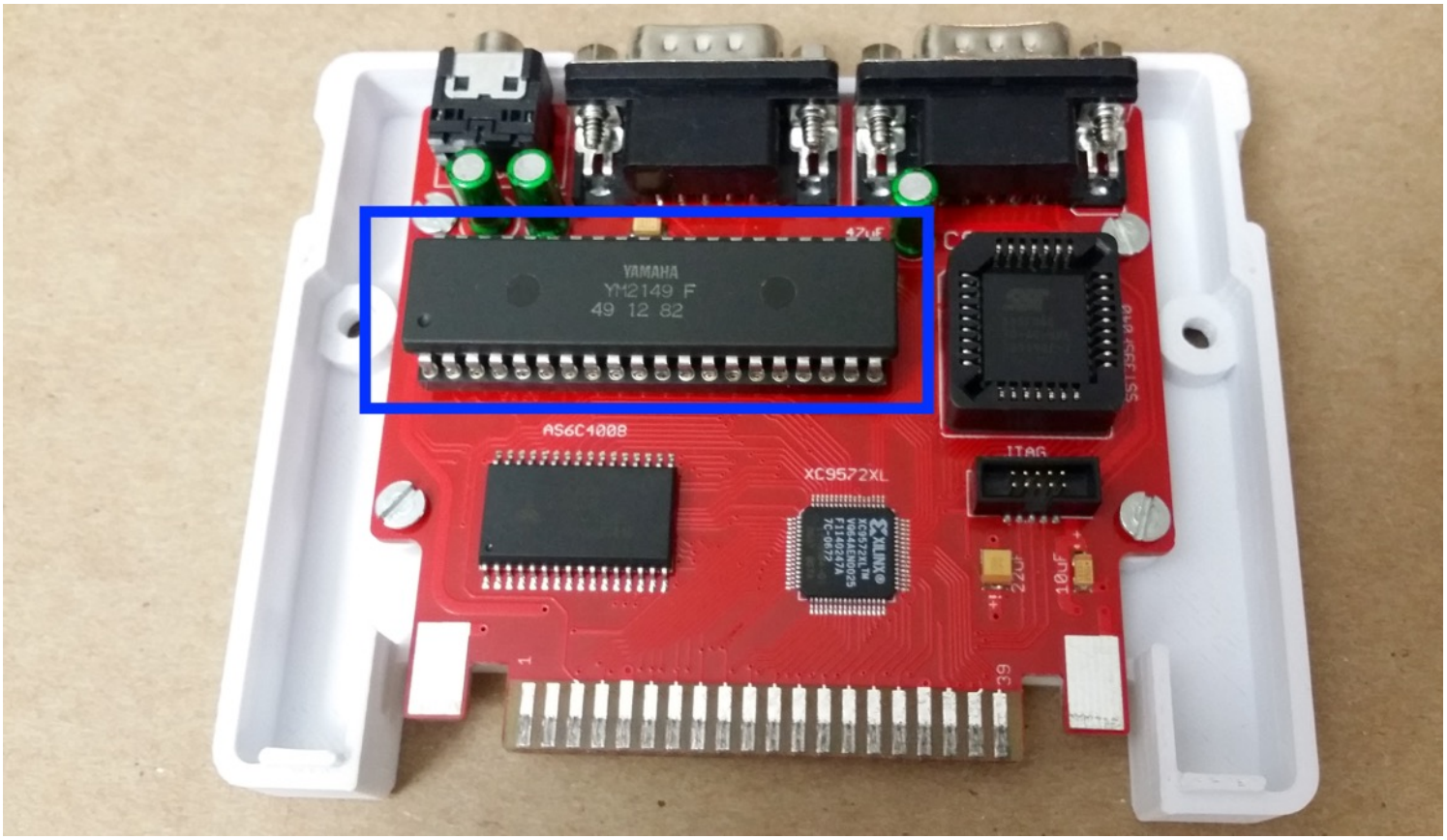
CoCo PSG and CoCo SDC in mini-MPI equipped CoCo2

Features include…

- YM2149 classic sound chip
- Digital joystick/controller ports
- 512K of RAM
- 512K of FLASH memory

YM2149 Sound Chip

The YM2149 included in the CoCo PSG is a classic sound chip from the 1980's that was used in the AtariST series and other systems. It is essentially a slightly modified version of the General Instruments AY-3-8910. Some of the differences include selectable clock divider, and double resolution/rate volume envelope table.



YM2149

This Yamaha chip, as interfaced in the CoCo PSG, has a software selectable master clock of either 2MHz or 1MHz allowing for a very wide range of frequencies to be generated. The two joystick ports are interfaced through the dual IO ports integrated into the YM2149.

Sound output from the CoCo PSG is available as line-out from the 1/8" jack on the cartridge itself, and is also routed internally to the CoCo via the sound-in line on the cart port allowing it to be output through RF modulators and built in video/sound boards.

512K SRAM

A single AS6C4008 IC adds RAM which is useable by programs needing additional memory.

512K FLASH

A single FLASH chip is provided for permanent storage of software and data on the cartridge.

Interface Logic

Tying these components together at the heart of the CoCo PSG is a Xilinx XC9572XL CPLD. This chip contains the necessary interface logic to connect the cartridge to the computer system, and provides for software control of the various features.

Programming Information

The CoCo PSG system adds the following addresses to the CoCo for accessing it's features;

ADDRESS	FUNCTION
$FF5A	MEMORY BANK 0 REGISTER
$FF5B	MEMORY BANK 1 REGISTER
$FF5D	CONTROL REGISTER
$FF5E	YM REGISTER SELECT PORT
$FF5F	YM DATA PORT

At system start-up, all registers are set to a default value of zero.

Access to all of these CoCo PSG registers are gated with *SCS, so when using in an MPI, *SCS must be switched to the slot with the CoCo PSG in order to write them. The registers will also respond to their respective addresses if bit 3 in the control register is set.

Memory reads from the FLASH and onboard RAM are gated with *CTS for reads. Bit 3 in the control register must be set in order to write the RAM, as *CTS is not active for writes.

Memory Registers

The PSG RAM and FLASH memory is accessable by the CoCo in the memory range of $C000-$FEFF. There are two banks, Bank 0 from $C000-$DFFF, and Bank 1 from $E000-$FEFF.

The 512K of memory in the FLASH and SRAM chips is divided into 64 - 8K segments (per chip). Any segment of RAM or FLASH can be presented in either memory bank by writing the memory bank registers. A program simply writes the memory segment number to the register controlling the bank where the memory will be accessed.

CHIP	SEGMENTS
FLASH	0-63
SRAM	128-191

The Control Register

The control register bits are used for the following functions;

BIT	FUNCTION
0	YM2149 MASTER CLOCK; 0=2MHz, 1=1MHz
1	Gameport A SEL signal (pin 7 of controller port A)
2	Gameport B SEL signal (pin 7 of controller port B)
3	Write Enable (for FLASH/SRAM); 0=disabled, 1=enabled
4	Autostart enable; 0=enabled, 1=disabled
5	FLASH programming enable; 0=disabled, 1=enabled
6	not used
7	not used

Using Banked Memory (SRAM and FLASH)

The extended memory provided by the CoCo PSG is designed to be used while the machine is in RAM/ROM mode. This is the mode that a CoCo1/2 will normally boot in, but a CoCo3 will have to be put into RAM/ROM mode (16K ROM) as it will normally boot into all-RAM mode after copying BASIC into RAM.

In order to read the contents of a memory segment, it must be mapped to one of the 8K banks in (16K) cartridge ROM space. This is done, as stated previously, by writing the desired segment number to one of the two memory registers. The last 256 bytes of memory in a segment are not available if mapped to the second bank ($E000-$FEFF) due to those memory locations being occupied by the CoCo's I/O space. They can be accessed by mapping the segment to the first memory bank at $C000-$DFFF.

Using the Extended RAM

As stated above, 8K segments of PSG extended RAM can be mapped to either bank in the 16K cartridge ROM space. To read the contents of an SRAM bank, nothing special has to be done, as reads are gated with *CTS just as normal ROM would be.

To write to the extended RAM, bit 3 in the control register must be set. This alerts the PSG to respond to write requests to cartridge ROM space, and also enables *SLENB for each write. *SLENB also enables the data buffer on MPIs for the writes, which would otherwise not let the data though.

Here is a bit of assembly code illustrating bank switching and writing the RAM. This simple RAM test the memory segment number to every byte in each segment, and checks that the value was successfully written.

```
287
288
289                                     ;   BANK TEST FIRST
290                                     ;   WRITE EACH SRAM BANK FULL OF THE
291                                     ;   VALUE FOR IT'S BANK (128-191)
292                                     ;   THEN SWITCH THROUGH THE BANKS,
293                                     ;   CHECKING THE VALUE RETAINED THERE.
294
295
296                 LDB     #%00011000  ;   SLENB ON, AUTOSTART OFF
297                 STB     CONTROL_REG
298
299                 LDA     #128        ;   FIRST RAM BANK VALUE
300
301 MORE8          STA     BANK0REG    ;   SET BANK 0
302                 LDX     #$C000      ;   POINT TO START OF BANK
303
304 MORE7          STA     ,X+
305                 CMPX    #$E000      ;   DONE FILLING BANK?
306                 BNE     MORE7       ;   NO, LOOP
307
308                 INCA                ;   NEXT BANK VALUE
309                 CMPA    #192        ;   ALL DONE HERE?
310                 BNE     MORE8
311
312                 LDB     #%00010000  ;   SLENB OFF, AUTOSTART OFF
313                 STB     CONTROL_REG
314
315                                     ;   BANKS FILLED, NOW PAGE THROUGH
316                                     ;   CHECKING FIRST VALUE IN EACH BANK
317                                     ;   USE BOTH BANK LOCATIONS FOR A
318                                     ;   BETTER TEST
319
320                 LDA     #128        ;   FIRST BANK VALUE
321 MORE9          STA     BANK0REG
322                 CMPA    $C000       ;   SAME?
323                 LBNE    BANK_ERROR  ;   GO REPORT ERROR, HALT TEST
324
325                 INCA
326                 CMPA    #192        ;   LAST BANK CHECKED?
327                 BNE     MORE9
328
```

Programming the FLASH Chip

The 512K FLASH chip on the CoCo PSG can be programmed directly from the CoCo. The computer must be in RAM/ROM mode, and the appropriate bits set in the control register as needed during programming. The FLASH used in the CoCo PSG is a SST39SF040 from Microchip Technology. The device data sheet may come in handy while writing your programming routines.

To send commands to the FLASH chip, two addresses are used (in the flash) that you will see reference to throughout the data sheet. These are $5555 and $2AAA. To simplify accessing these addresses in the FLASH chip while also accessing the memory segments in cartridge ROM space, these are mapped to CoCo addresses $BFFF and $BAAA respectively, but only when control register bit 5 (FLASH Programming Enable) is set. Bit 3 enabling writes to cart space must also be set. The CoCo's internal memory will also receive these writes, but since the machine must be in RAM/ROM mode anyway, it's of no consequence.

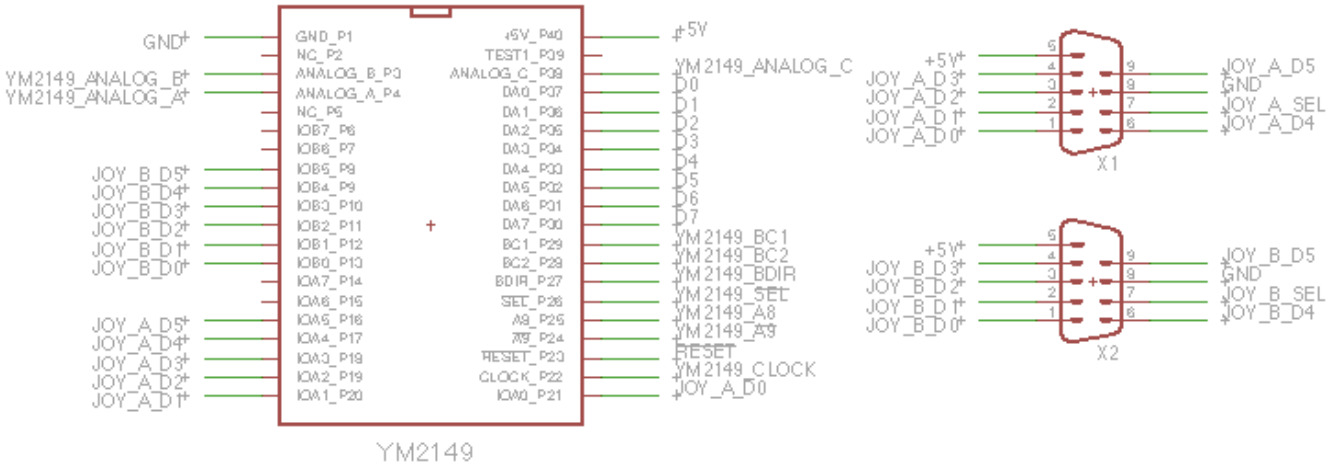
The following example code erases the entire FLASH chip in the CoCo PSG, and illustrates accessing FLASH programming features. For additional FLASH chip commands, including programming the chip, please refer to the manufacturer data sheet.

```
110
111 FLASH5555         EQU     $B555         ;   WRITES HERE WHEN CONTROL REG BIT 5 IS SET GO
112                                         ;   TO FLASH CHIP ADDRESS $5555
113 FLASH2AAA         EQU     $BAAA         ;   WRITES HERE WHEN CONTROL REG BIT 5 IS SET GO
114                                         ;   TO FLASH CHIP $2AAA
```

```
269
270 ;**************************************************************************************
271 ;   ERASEFLASH - ERASES THE FLASH CHIP
272 ;
273 ;**************************************************************************************
274
275 ERASEFLASH  JSR     SWITCHPSG       ;   SWITCH SCS TO PSG / SLENB ENABLED
276
277             LDA     $FF5D           ;   GET PSG CONTROL REG VALUE
278             ORA     #%00100000      ;   ENABLE FLASH PROGRAMMING (BIT 5)
279             STA     $FF5D           ;   SET IT
280
281
282                                     ;   LET'S ERASE ENTIRE CHIP, SEQUENCE...
283                                     ;
284                                     ;   WRITE $5555 WITH $AA
285                                     ;   WRITE $2AAA WITH $55
286                                     ;   WRITE $5555 WITH $80
287                                     ;   WRITE $5555 WITH $AA
288                                     ;   WRITE $2AAA WITH $55
289                                     ;   WRITE $5555 WITH $10
290                                     ;   CHECK TOGGLE BIT BEFORE CONTINUING
291
292             LDD     #$AA55          ;   LOAD FIRST 2 BYTES OF ERASE SEQUENCE
293             STA     FLASH5555       ;   FIRST BYTE TO $5555
294             STB     FLASH2AAA       ;   SECOND BYTE TO $2AAA
295             LDD     #$80AA          ;   LOAD NEXT TWO BYTES OF ERASE SEQUENCE
296             STA     FLASH5555       ;   THIRD BYTE TO $5555
297             STB     FLASH5555       ;   FOURTH BYTE TO $5555
298             LDD     #$5510          ;   LOAD NEXT TWO BYTES OF ERASE SEQUENCE
299             STA     FLASH2AAA       ;   FIFTH BYTE TO $2AAA
300             STB     FLASH5555       ;   FINAL BYTE TO $5555
301
302
303             LDA     $FF5D           ;   GET PSG CONTROL REG VALUE
304             ANDA    #%11011111      ;   DISABLE FLASH PROGRAMMING (BIT 5)
305             STA     $FF5D           ;   SET IT
306
307
308                                     ;   100,000 CYCLE WAIT
309                                     ;   SLIGHTLY MORE THAN THE 100MS ERASE CHIP TAKES
310
311             LDX     #8334           ;   FOR 100,000 CYCLES
312 WLOO        LEAX    -1,X            ;   DEC COUNTER     [5]
313             CMPX    #0              ;   ZERO?           [4]
314             BNE     WLOO            ;   NO, LOOP...     [3] [12] = 12 CYCLES PER LOOP
315
316             JSR     SWITCHSDC       ;   BACK TO ESTABLISHED STATE (SDC/NO SLENB)
317
318             RTS                     ;   RETURN TO CALLING ROUTINE
319
320 ;   END OF ERASE FLASH ROUTINE
321
322
```

Accessing the Joystick Ports

COCO PSG - GAMEPORT WIRING



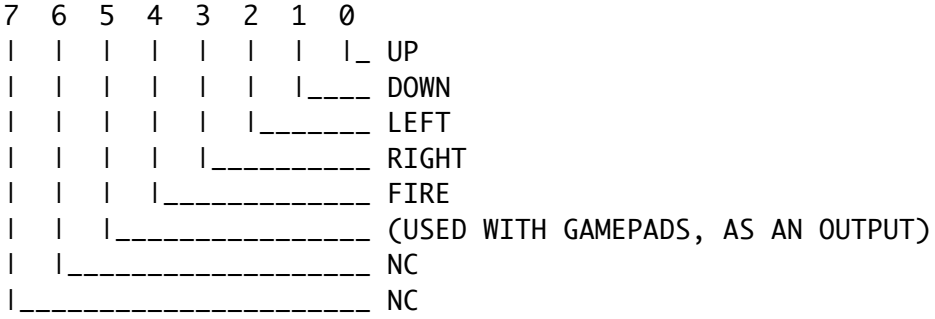
POWER, CPLD, OR

YM IO PORT PIN	9-PIN CONNECTOR	GAMEPORT
PORT A, A0	PIN 1	A
PORT A, A1	PIN 2	A
PORT A, A2	PIN 3	A
PORT A, A3	PIN 4	A
PORT A, A4	PIN 6	A
PORT A, A5	PIN 9	A
PORT A, A6	NC	
PORT A, A7	NC	
+5V	PIN 5	A
GND	PIN 8	A
CPLD, JOY_A_SEL	PIN 7	A
PORT B, A0	PIN 1	B
PORT B, A1	PIN 2	B
PORT B, A2	PIN 3	B
PORT B, A3	PIN 4	B
PORT B, A4	PIN 6	B
PORT B, A5	PIN 9	B
PORT B, A6	NC	
PORT B, A7	NC	
+5V	PIN 5	B
GND	PIN 8	B
CPLD, JOY_B_SEL	PIN 7	B

READING THE GAME PORTS THROUGH THE YM CHIP

FIRST, SET THE IO REGISTERS TO INPUT (REG7, BITS 6-7)
THEN READ THE REGISTER FOR THE GAME PORT (14 OR 15, A AND B RESP)
SWITCHES ARE ACTIVE LOW (GROUNDED)

VALUE FROM YM2149 REGISTER (8-BITS) - BASIC ATARI STICK



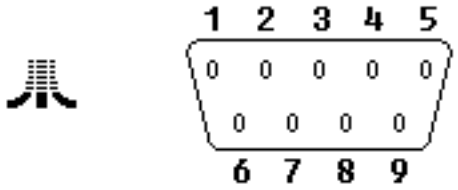
POSSIBLE VALUES FOR A STANDARD ATARI STICK (4 DIRECTIONAL, 1 BUTTON)

1111 1110 = UP, NO FIRE
1111 1101 = DOWN, NO FIRE
1111 1011 = LEFT, NO FIRE
1111 0111 = RIGHT, NO FIRE

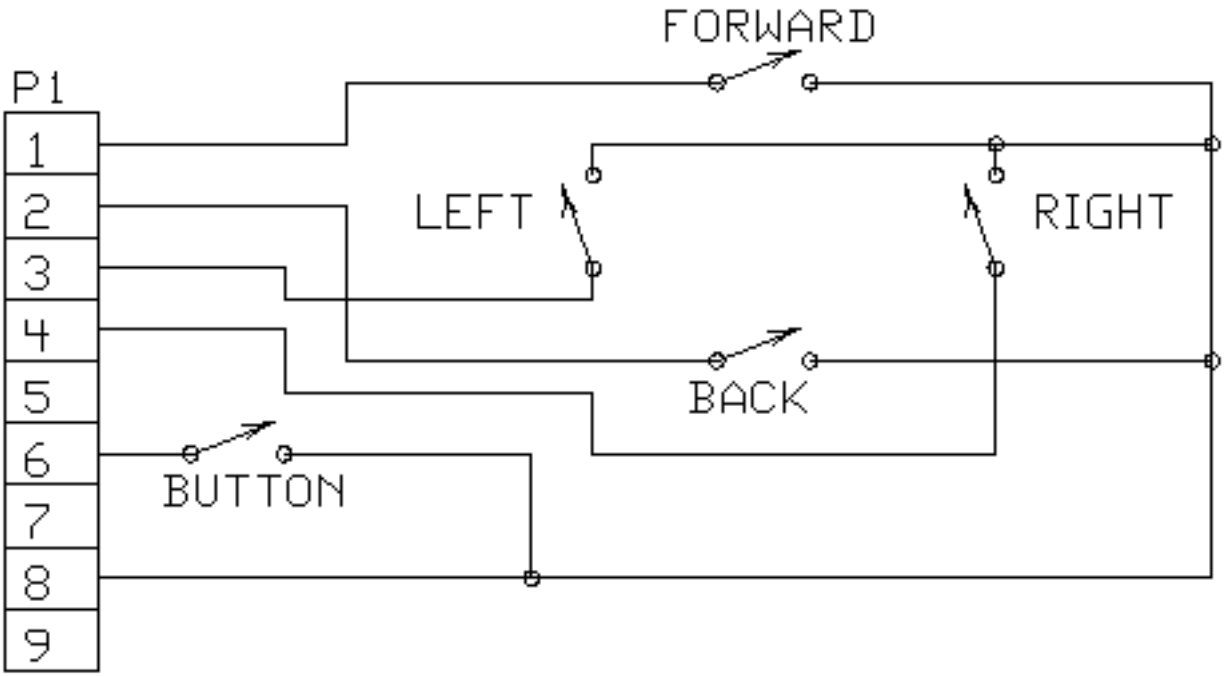
1111 0110 = UP/RIGHT, NO FIRE
1111 1010 = UP/LEFT, NO FIRE
1111 0101 = DOWN/RIGHT, NO FIRE
1111 1001 = DOWN/LEFT, NO FIRE

AND ALL THESE VARIATIONS WITH BIT 4 = 0, WHICH WOULD INDICATE FIRE BUTTON DOWN

EXAMPLE OF TYPICAL ATARI JOYSTICK PINOUTS & SWITCH CIRCUIT
(NOT COCO-PSG GAMEPORT PINOUT)



1. **Joystick Up**
2. **Joystick Down**
3. **Joystick Left**
4. **Joystick Right**
5. **B input Paddle/Touch Tablet**
6. **Input Trigger**
7. **+5V**
8. **Gnd**
9. **A input Paddle/Touch Tablet**



YM2149 Register Select and Data Port

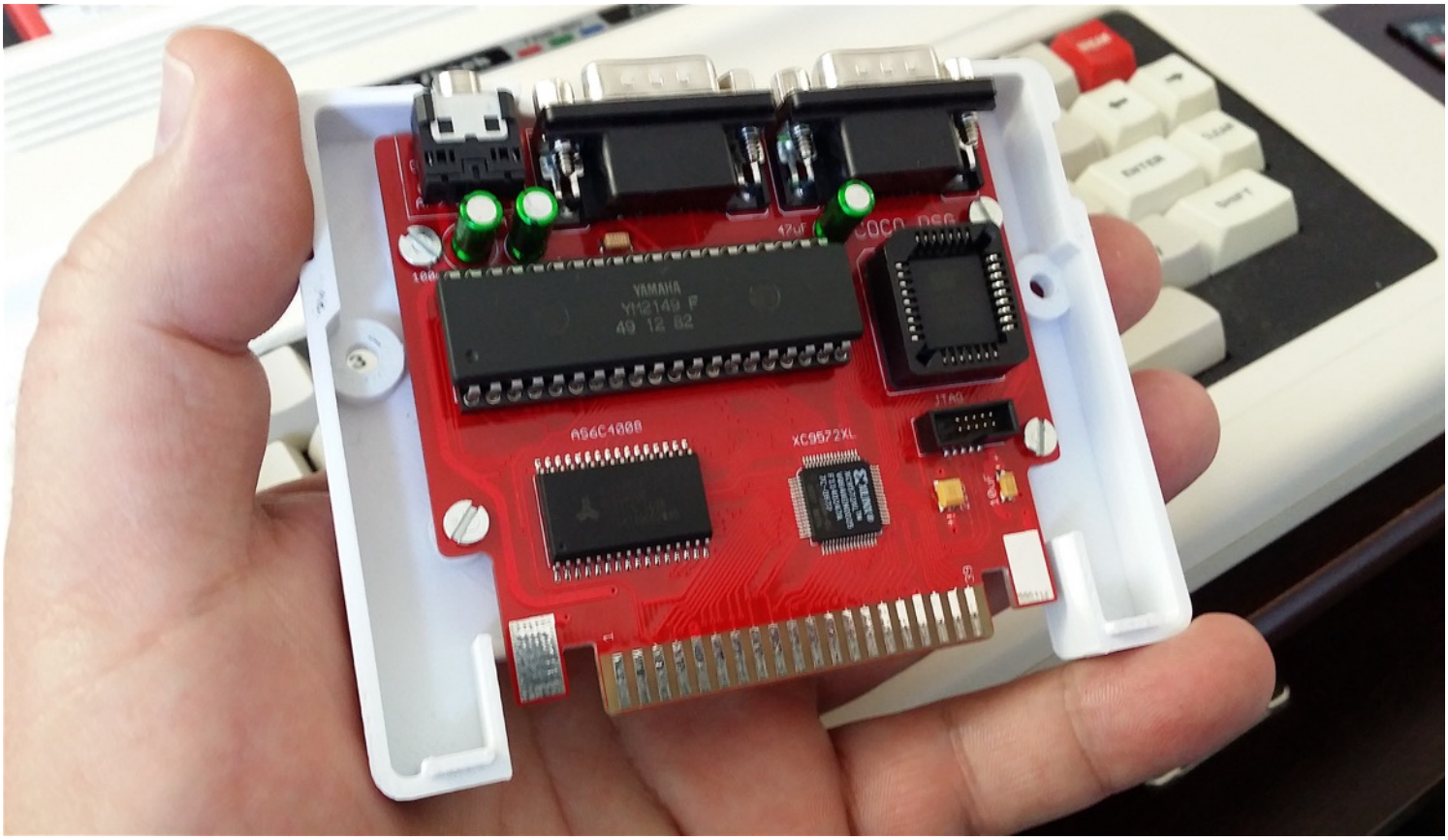
These two addresses allow for direct access of the YM's internal registers/data. Please refer to the YM2149 Datasheet for details on their usage.

YM2149 register select, $FF5E

– select the YM internal register to work with by writing it's number here

YM2149 data port, $FF5F

- read or write the YM internal register value here



Thank you for supporting CoCo hardware and software developers, and I hope you enjoy and get a lot use out of your new CoCo PSG!

- Ed Snider (AKA Zippster)