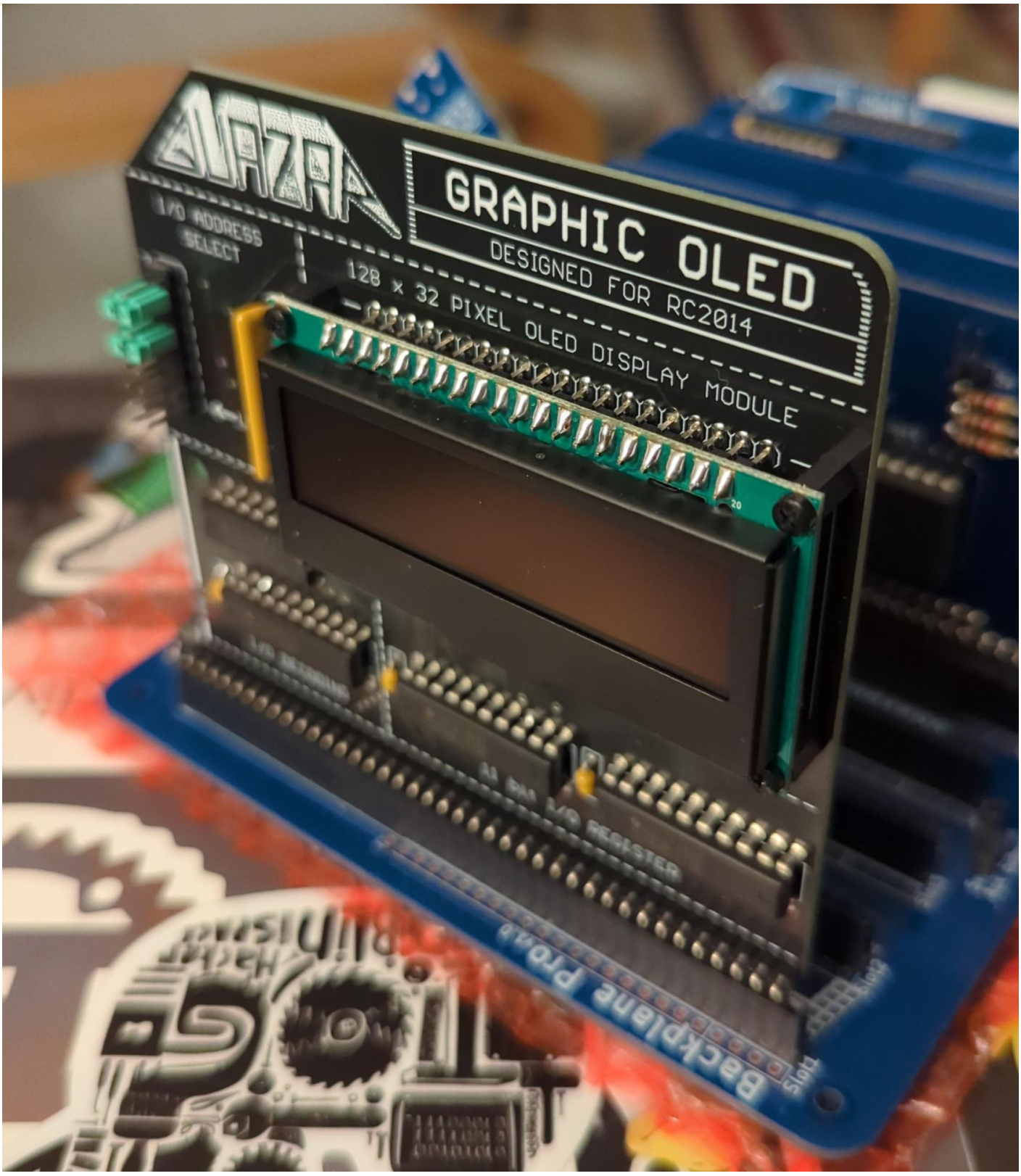
RC2014 Cards

RC2014 Cards Resources

- https://hackaday.io/project/159057-game-boards-for-rc2014
  - https://www.youtube.com/watch?v=1K_cc8wFURc
- TMS9918A based video card
- SN76489 based sound card
- https://www.tindie.com/stores/mfkamprath/
- ESP8266 Wifi Module For RC2014
- DS1302 Real Time Clock Module for RC2014

Quazar OLED interface



Official site: https://2014.samcoupe.com/#graphicoled

Based on SSD1305 OLED display (128x32 version?): SSD1305.pdf

Uses `0x50` I/O port by default (fully configurable). I/O decoder uses 74HCT02 NOR gates and 74GCT688 8-bit comparator.

The SSD1305 chip is placed behind two 74HCT573 8-bit latches so it can be timed slower than Z80 bus:

- One latch takes 8 bit data bus and forwards to the OLED chip.
- The other latch takes 3 control bits from higher address buss byte as output from B register.
- Latches take input on I/O bus write when base address matches.
  - Compactor is enabled when RW and IORQ are active (low).
  - If address matches the I/O address select the latches will copy and store data from data bus and first 3-bits from high address byte.
  - The latched data is presented to OLED until the next I/O write to the card occurs.

Software

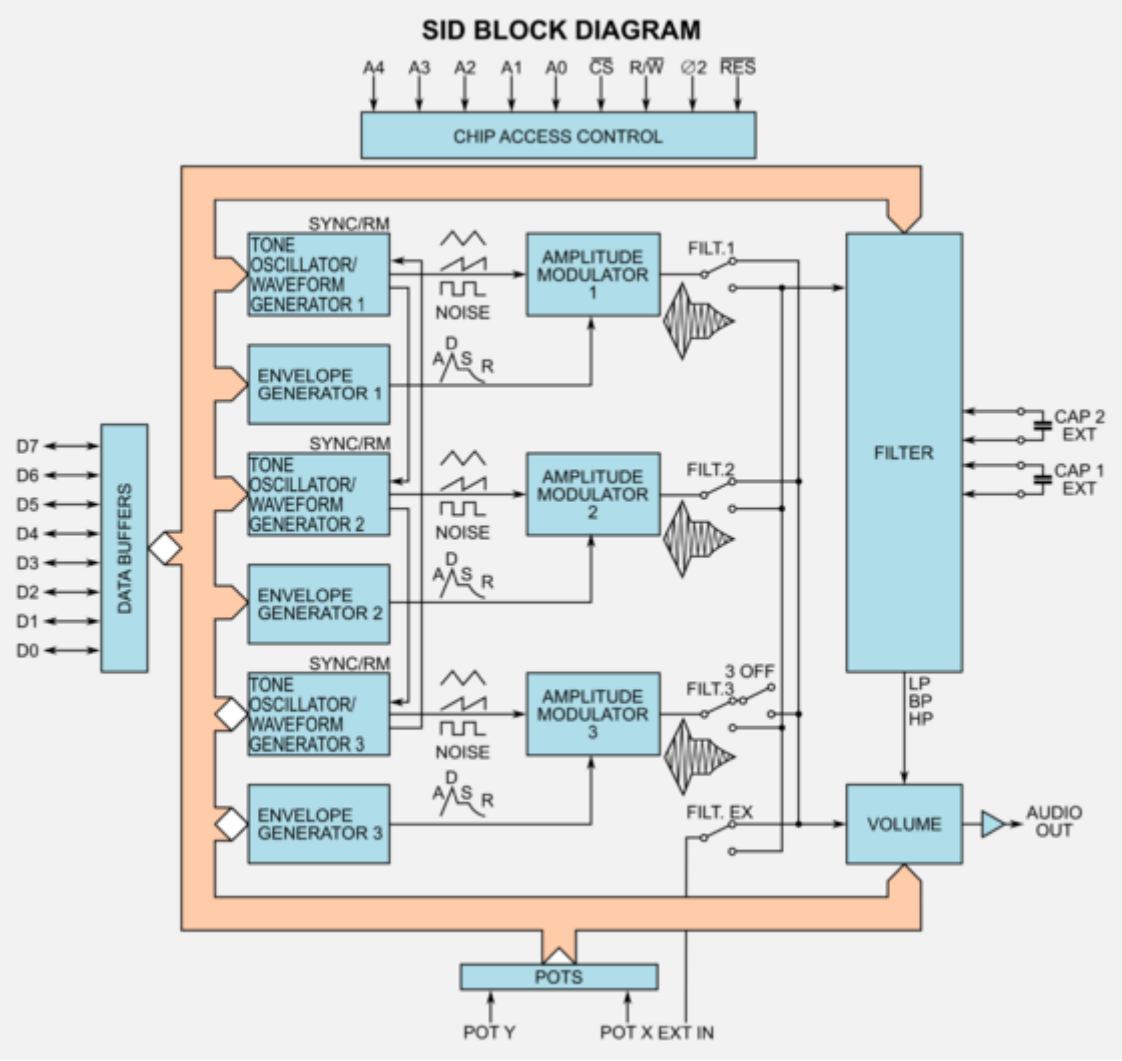
I have implemented a program to display images using instructions form provided manual:

- rc2014-oled - `oled` program for CP/M and PNG image converter.

Quazar SID Interface

SID Programming

- https://www.c64-wiki.com/wiki/SID
- MOS6581_SID_data_sheet.pdf



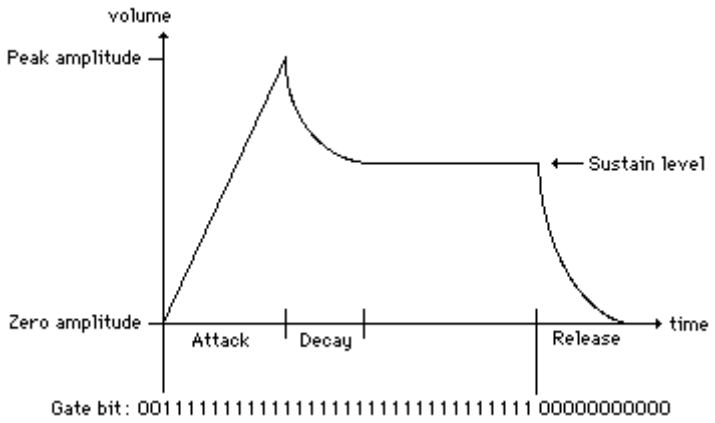
Memory Addresses of the SID

Reg	Function
0	frequency voice 1 low byte
1	frequency voice 1 high byte

2	pulse wave duty cycle voice 1 low byte							
	7..4				**3..0**			
3	—				pulse wave duty cycle voice 1 high byte			
4	control register voice 1							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	noise	pulse	sawtooth	triangle	test	ring modulation with voice 3	synchronize with voice 3	gate
	7..4				**3..0**			
5	attack duration				decay duration voice 1			
6	sustain level				release duration			
7	frequency voice 2 low byte							
8	frequency voice 2 high byte							
9	pulse wave duty cycle voice 2 low byte							
	7..4				**3..0**			
10	—				pulse wave duty cycle voice 2 high byte			
11	control register voice 2							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	noise	pulse	sawtooth	triangle	test	ring modulation with voice 1	synchronize with voice 1	gate
	7..4				**3..0**			
12	attack duration				decay duration voice 2			
13	sustain level				release duration voice 2			
14	frequency voice 3 low byte							
15	frequency voice 3 high byte							
16	pulse wave duty cycle voice 3 low byte							
	7..4				**3..0**			
17	—				pulse wave duty cycle voice 3 high byte			
18	control register voice 3							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

	noise	pulse	sawtooth	triangle	test	ring modulation with voice 2	synchronize with voice 2	gate
	7..4				**3..0**			
19	attack duration				decay duration voice 3			
20	sustain level				release duration voice 3			
21	—					filter cutoff frequency low byte		
22	filter cutoff frequency high byte							
23	filter resonance and routing							
	7..4				**3**	**2**	**1**	**0**
	filter resonance				external input	voice 3	voice 2	voice 1
24	filter mode and main volume control							
	7	**6**	**5**	**4**	**3..0**			
	mute voice 3	high pass	band pass	low pass	main volume			
25	paddle x value (read only) - not available							
26	paddle y value (read only) - not available							
27	oscillator voice 3 (read only) - not available							
28	envelope voice 3 (read only) - not available							

Envelopes



Value	Attack Rate	Decay/Release Rate

Decimal	(Time/Cycle)	(Time/Cycle)
0	2 mS	6 mS
1	8 mS	24 mS
2	16 mS	48 mS
3	24 mS	72 mS
4	38 mS	114 mS
5	56 mS	168 mS
6	68 mS	204 mS
7	80 mS	240 mS
8	100 mS	300 mS
9	250 mS	750 mS
10	500 mS	1.5 S
11	800 mS	2.4 S
12	1 S	3 S
13	3 S	9 S
14	5 S	15 S
15	8 S	24 S

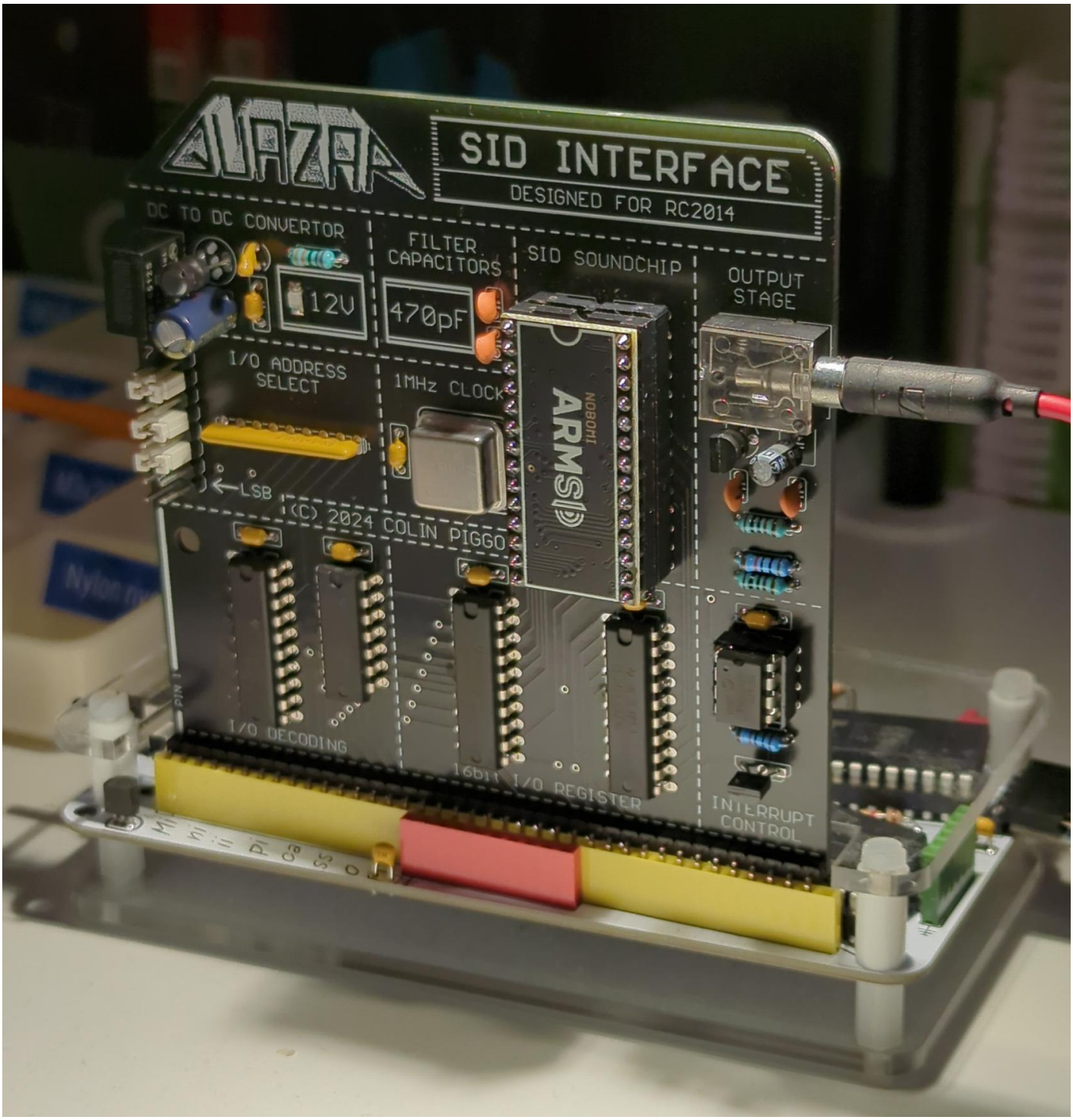
Frequency

```
\ Frequency table:
: SID:NOTE:C4  4389 ;
: SID:NOTE:C4# 4650 ;
: SID:NOTE:D4  4927 ;
: SID:NOTE:D4# 5220 ;
: SID:NOTE:E4  5530 ;
: SID:NOTE:F4  5859 ;
```

```
: SID:NOTE:F4# 6207 ;
: SID:NOTE:G4  6577 ;
: SID:NOTE:G4# 6968 ;
: SID:NOTE:A4  7382 ;
: SID:NOTE:A4# 7821 ;
: SID:NOTE:B4  8286 ;

\ Octave shifts
: SID:OCT:UP 2* ;
: SID:OCT:DOWN 2/ ;
```

Using with RC2014 Mini II Picasso



Limitations

- Interrupt timer won't work since first half of memory is ROM and can't install interrupt handler. This will work with CP/M cards where all memory space is backed by RAM.
- It is not possible to read data from SID chip. Paddled and voice 3 data and ADSR registers cannot be read.

Basic

Set ROM address to `0 100` (just *Bank 2* set) to boot into *Microsoft BASIC (Phil Green)*.

Insert SID interface card: marked `A15` pin is pin 1.

Use the following test BASIC program (as provided in the instructions):

```
10 DATA 205,7,10,123,66,14,84,237,121
20 DATA 0,0,0,203,248,237,121,195,125,17
30 FOR A=-1024 TO -1006
40 READ D
50 POKE A,D
60 NEXT A
70 POKE -32695,0
80 POKE -32694,252
90 LET A=USR(1024)
100 LET A=USR(6159)
101 LET A=USR(1280)
102 LET A=USR(1776)
103 LET A=USR(1041)
110 FOR D=256 TO 511
120 LET A=USR(D)
130 NEXT D
140 GOTO 110
```

Type `RUN` to start it. You should hear "siren" audio effect.

CamelForth

Set ROM address to `0 001` (just *Bank 8* set) to boot into *CamelForth (Justin Skists)*.

Paste following Forth program.

```
\ PS2!     c c p-addr --   Data byte (A), high address byte (B), port address (C) - OUT to
port with A and B registers set
: PC2! SWAP >< OR PC! ;
\ SID!     c c --          Data value, register (0-31); bits 5,6 are for interrupt settings;
bit 7 (/CS) is handled
: SID!
  2DUP 128 OR 84 PC2!
  2DUP 127 AND 84 PC2!
  128 OR 84 PC2! ;
```

```
: SIDRST 0 24 SID! 0 4 SID! ;
: SIDTEST 15 24 SID! 10 0 DO 255 0 DO I 1 SID! LOOP LOOP 0 24 SID! ;

SIDRST
15 24 SID!
0 5 SID!
240 6 SID!
17 4 SID!
SIDTEST
```

Decimals used:

- 84 = 0x52 - I/O port number
- 24 = 0x18 register 0x18 (24), no interrupt
- 128 - high bit set
- 127 - all but high bit set

See project page for CamelForth SID library: sid.4th