How do you port conio to another target? #1683

Answered by mrdudz **ProxyPlayerHD** asked this question in **Q&A**

ProxyPlayerHD on Feb 20, 2022

specifically the Console related functions, as the only hardware dependant thing they should require are:
a function to print a character, and a function to get a character.

i already got both for my target, example:

```
void output_char(char value){
        while (*((uint8_t*)0xF701) & 1 == 1){};         // Constantly check the Status bit 0 of the
UART
        *((uint8_t*)0xF700) = value;                    // If it's 0, write the byte to the UART
}

char input_char_wait(){
        while ((*((uint8_t*)0xF701)) >> 1 & 1 == 1){};  // Constantly check the Status bit 1 of the
UART
        return *((uint8_t*)0xF700);                     // If it's 0, read a byte from the UART
}
```

but i've not be able to figure out how to actually pull this off as the library seems so intertwined with cc65 that i don't know which source files are required to compile/assemble the conio library and which aren't.
I was also not able to find any useful documentation for this online either.

so how are you supposed to do it? given these 2 base functions how do i port atleast `cprintf` to a new target system?

↑ 1

Answered by **mrdudz** on Feb 20, 2022

What you describe applies more to the stdio things than to conio :) conio is ment for fast and efficient screen output (and input) and provides a bunch of related features, like placing the cursor or changing colors - and all of this is platform specific.

What kind of platform is this? Are you sure conio is even what you want?

13 comments

Oldest | Newest | Top

mrdudz on Feb 20, 2022 Collaborator

What you describe applies more to the stdio things than to conio :) conio is ment for fast and efficient screen output (and input) and provides a bunch of related features, like placing the cursor or changing colors - and all of this is platform specific.

What kind of platform is this? Are you sure conio is even what you want?

Marked as answer | ↑ 1 | ☺ | 0 replies

Write a reply

Answer selected by **ProxyPlayerHD**

ProxyPlayerHD on Feb 20, 2022 Author

The platform is my own 65C02 SBC, 61.75kB of RAM, 2kB of Boot ROM, and 256B of IO.
and besides Serial (in form of an FT240X) it has no other IO functionality.

i've already made my own Serial IO library that allows me to print characters, strings, and decimal/hex/fixed point numbers.
but it's missing something `printf` compatible, so i was hoping that cc65 had some existing `printf` function that i could simply copy and port by replacing it's character output function.

↑ 1 | ☺ | 0 replies

Write a reply

mrdudz on Feb 20, 2022 Collaborator

I'd start with the lowlevel _read and _write functions, which stdio is based on, that will make the usual printf and getchar work

↑ 1 | ☺ | 0 replies

ProxyPlayerHD on Feb 20, 2022 Author edited

alright i can do that, i found a few versions of `read.s` and `write.s` for various systems.
i'll use sym1's as a starting point to write my own version.
it looks pretty staight forward... as it seems like i only really need to change the outch and getch functions.

but what after that? i can assemble the files and inlcude them while linking, is that all that's needed to make it use my custom versions of _read and _write? do i have to modify something in the stdio header file as well?

↑ 1 ☺ 0 replies

Write a reply

mrdudz on Feb 20, 2022 Collaborator

yeah just link them, that should be enough to get you started

↑ 1 ☺ 0 replies

Write a reply

ProxyPlayerHD on Feb 20, 2022 Author edited

alright it actually worked! that was surprisingly simple, i feel like that should be included in the "how to create a custom target for cc65" tutorial. just because having access to printf when starting to program for cc65 is a pretty important thing.

anyways, i now got CoreMark ported to cc65 (no idea if someone already did that).
but with no timer on the SBC i sadly can't run the full benchmark until i make a little addon card that has a 65C22 or something on it. (currently it just says that the benchmark needs to run for atleast 10s to be valid, and since i made `clock()` just return 0 it obviously won't work)
when i actually get it running i can post some results if you're interested.

↑ 1 ☺ 0 replies

Write a reply

ProxyPlayerHD on Feb 20, 2022 Author edited

hmm, i've noticed an oddity, whenever i print a 32 bit value it only prints out the upper 16 bits.

so for example:
`printf("%X\n", 0xDEADBEEF);` outputs `DEAD` instead of `DEADBEEF`
when i try it in decimal with the same value it outputs `-8531` which is `0xFFFFDEAD` in hex

so somewhere in the printf function it shifts the whole upper word into the lower, and then sign extends it.

↑ 1 ☺ 0 replies

Write a reply

mrdudz on Feb 20, 2022 Collaborator

%x expects an int, so 16bit - try %lx

↑ 1 ☺ 0 replies

Write a reply

ProxyPlayerHD on Feb 20, 2022 Author

ah, that works!
this also explains why CoreMark was unable to display the time in seconds, as it just used "%d" instead of "%ld".
but it's still confusing that it would take the UPPER half of the 32-bit number instead of the lower half.

↑ 1 ☺ 0 replies

Write a reply

mrdudz on Feb 20, 2022 Collaborator

(i am moving this to discussions)

↑ 1 ☺ 0 replies

Write a reply

 **ProxyPlayerHD** on Feb 20, 2022 **Author**

so it's just c being c, got it.
also oooh, fancy discussions! i haven't used this feature of github yet.

anyways i got it working now, and i learned that porting stdio is pretty straight forward (assuming you ignore file specific things like fopen, fclose, etc).
so i guess i can just close this now.

in case you were wondering on the CoreMark results:

```
2K performance run parameters for coremark.
CoreMark Size    : 666
Total ticks      : 4239123239
Total time (secs): 211        (manually calculated: 211.956162 sec)
Iterations/Sec   : 0          (manually calculated: 0.235897836 iterations/sec)
Iterations       : 50
Compiler version : cc65 V2.19
Compiler flags   : -t none -O --cpu 65C02
Memory location  : MALLOC
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x0158
Correct operation validated. See README.md for run and reporting rules.
```

so the total score is 0.236 CoreMark, or 0.0118 CoreMark/MHz (an ARM Cortex-M0 has a score of ~2.33 CoreMark/MHz)

↑ 1 ☺ 0 replies

Write a reply

 **polluks** on Feb 21, 2022

At least we have https://cc65.github.io/doc/customizing.html
Feel free to add a stdio chapter :-)

↑ 1 ☺ 0 replies

Write a reply

 **greg-king5** on Feb 23, 2022 Collaborator

conio is smaller than stdio. If you ever want to use the conio library, here are the three base functions that you need:

```
#include <conio.h>

// UART registers
struct __FT240X {
    unsigned char data;
    unsigned char status;
};
#define SERIAL (*(struct __FT240X *)0xF700)

void __fastcall__ cputc(char c)
{
    do {} while ((SERIAL.status & 0x01) != 0);  // Wait until the send register is empty.
    SERIAL.data = c;
}

unsigned char kbhit(void)
{
    return (SERIAL.status & 0x02) == 0;  // True if a received byte is ready.
}

char cgetc(void)
{
    do {} while (!kbhit());  // Wait until a received byte is ready.
    return SERIAL.data;
}
```

↑ 1 ☺ 0 replies

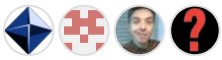
Write a reply

Q&A

Labels

question libs

4 participants



Converted from issue

This discussion was converted from issue #1682 on February 20, 2022 20:05.

Events

ProxyPlayerHD Marked an Answer 1y