

%%% ABCbus V0.6 %%%

Aug 7, 2016.

The **ABCbus interface** is a general memory emulation and peripheral interface between an old Z80 computer and a PIC32 microcontroller. It provides ROM and RAM emulation, disk emulation from an SD card, WiFi and extra serial port interface for the Swedish computer ABC80. It should be easy to adapt for other Z80 architectures and other architectures could be supported with minor changes.



The green pcb is the interface between the ABC80/800 bus and the PIC32 card UBW32. A WiFly module is also mounted as well as a SD card holder (on other side). The USB-mini connector in the upper left of the red UBW32 card can be used as a virtual com port. (The vertical USB-A is not implemented).

The project was developed by Robert Juhasz and Anders Jansson in 2012/13.

The www.abc80.org mailing list is a good forum for discussions and bug reports.



Features

There is a micro-SD card holder on the interface card. The SD card should be max 2 GB formatted with FAT16 (only single partitions work?).

ABC80 can load/save files directly on the SD card. It can also use an image of an authentic floppy or hard drive.

The SD card is on the ABC80 side of the board and can easily be popped in or out. All other features are located on the back side as seen in the picture below.

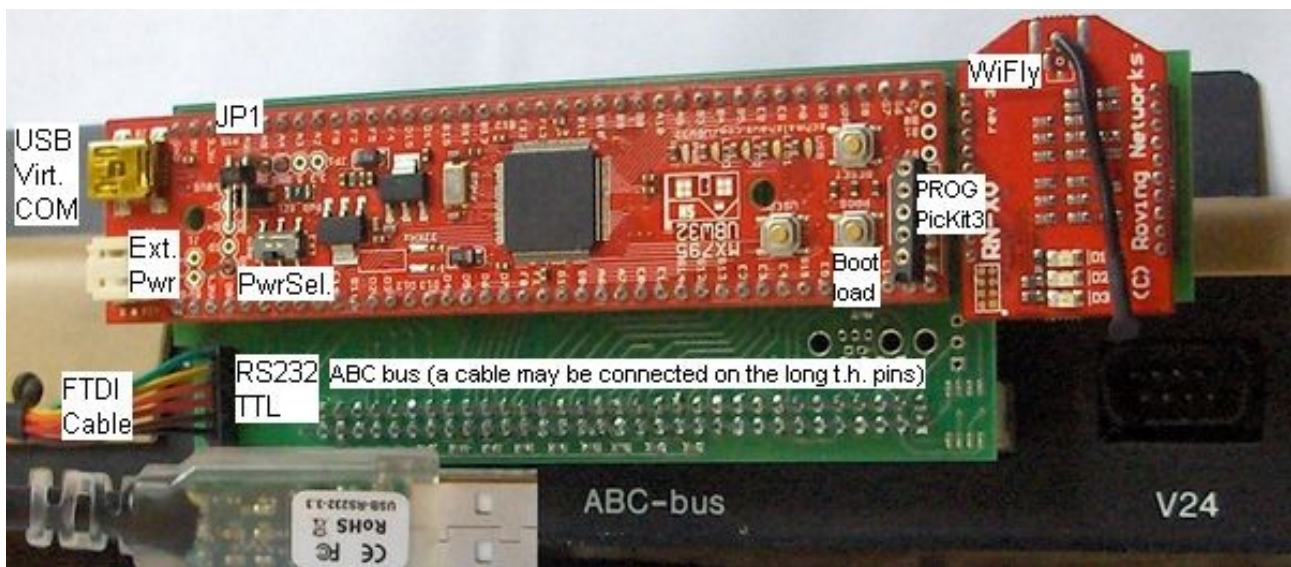


Fig. (The WiFi, JP1, PROG connector, FTDI cable and long bus pins are optional)

The WiFi and RS232 TTL port (optionally USB com port) are currently supported as memory mapped devices with 256 byte Tx and Rx buffers (see abcMemory.h). The TTL port uses 3.3 V logic and can be used with a 3v3 FTDI cable.

There are 5 LEDs on the UBW32 board.

- Blue is always ON (power),
- Green shows that the SD card is ready
- Yellow indicates an SD card read/write operation.
- White is ON during boot and while the ABC80 reset-button is active.
- Red indicates a critical PIC32 exception error and will not be seen anymore (we hope :-)

Disk emulation

En ABC80 hårddisk emuleras med en Winchester disk image. Filen abcdisk.hd0 ger 4 hårddiskar med 10 MB vardera.

Vid power ON / reset körs filen
HD0:START80.BAS.

Dysken innehåller en massa program i olika UFD-bibliotek. UFD-DOS finns beskrivet i bitförbit del 1 (länk saknas).

Här är ett exempel.

BYE

UFD

hugo (går till hugo.ufd på HD2)

exit

run highrez

```
UFD-DOS ver. 20
DR_: motsvaras av HD_:
Gammal adressering

ABC80
;call(16384)
 0   %% ABCbus Interface 0.6 %%
ABC80
█
```

För att skriva/läsa disk image i en PC finns abcwrite/abcread av Hans Peter Anwin. Kör programmen utan parameter för att få en beskrivning. Disk image kan användas i emulatorn ABCWin2.

SD0:

Ett specialskrivet device SD0: kan ladda och spara program direkt på SD-kortet.

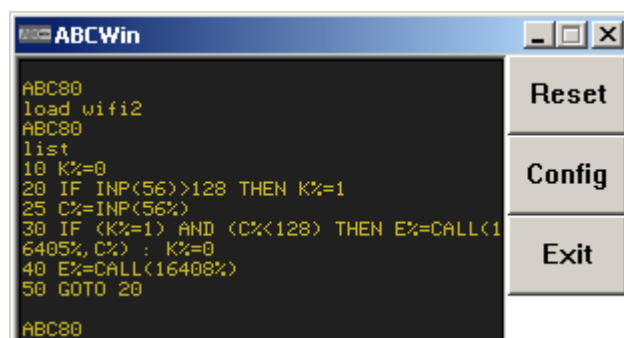
On the ABC80 side, the device **SD0:** can be installed as the default storage device by **;CALL(16384)**. A simple **LOAD TED** should then work if **TED.BAC** is present in the root directory of the SD card. A number of programs have been tested such as **TED.BAC** and **ASS.BAC** editor/assembler suite, **RULLE.BAS** (speech synthesis), **Lökaren.32k**, **PINBALL.BAS**, etc. Files can be copied to the SD card in a PC using a card reader. Programs can be downloaded from <http://www.devili.iki.fi/pub/Luxor/software/ABC-klubben/abc80/>

```
ABC80
;call(16384)
 0   %% ABCbus Interface 0.6 %%
ABC80
load lib2
ABC80
list
10  A=0
20  OPEN "cmd:" ASFILE 1
30  PRINT #1,"LIB";
40  ONERRORGOTO 70
50  INPUTLINE #1,A#
60  PRINT A#;
70  A#A#1 : IF A#>15 THEN A#0 : GET A#
80  GOTO 40
90  CLOSE 1
ABC80
```

PRINT call(16384) installerar också CMD: för särskilt implementerade kommandon till PIC32. För närvarande är det mest kommandot "LIB" som kan användas för att lista filer i SD0.

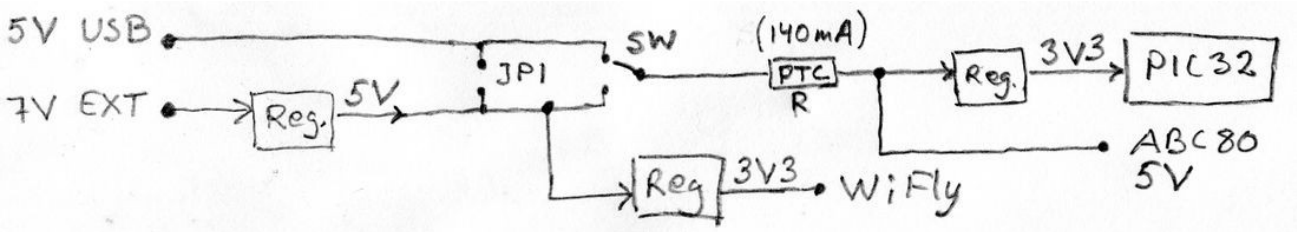
Initieringen med ;call(16384) kan förstås göras i START80.BAS om man vill. Både SD0 och CMD är för närvarande minnesmappade och fungerar alltså inte med ABC800.

SD0 BAC-filer är kompatibla med ABCWin1, men inte BAS-filer.



Power

With the UBW32 power switch in the ext position, the PIC32 and SD card is powered from the ABC80 bus. The usb position is useful for updating the firmware via USB when the unit is disconnected from ABC80. It may also be used for powering a USB device from the UBW32, e.g. a mouse/keyboard. It should not draw much power though since the ABC80 bus has a rather low power limit. There is a PTC resistor on UBW32 that limits the current to about 140 mA.



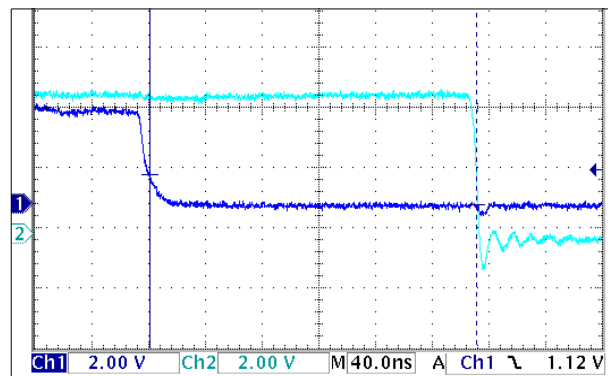
Connecting the USB cable to PC while connected to ABC80, 5V from the ABC80 will be connected to USB 5V from USB via the PTC resistor. It can be noted that the LEDs go off 1-2 seconds later than when the switch is in the EXT position. This does not appear to cause any harm.

The WiFi module is preferably powered by external power (ca 7V 0.3A) via a 5V regulator on the UBW32 and a separate 3V3 regulator on the ABCbus card. It is possible to drive the WiFi from USB by mounting a jumper JP1. Power från ABC80 via PTC resistor kan också fungera?

Technology

The PIC32 microcontroller is connected directly to the system bus of the ABC80. The /RD and /WR lines are connected to PIC32 interrupts, emulating RAM and ROM from 64k internal PIC ram memory. The memory is defined in blocks of 256 bytes, emulated as either read/write or read only memory.

The PIC32 interrupt routines uses highly optimized assembler to keep up with the M1 cycle of a 3 MHz Z80. All bus interrupts use highest interrupt priority without any context saving, and no other interrupts must be allowed to interfere. Data is output around 230 ns after the RD# strobe goes low.

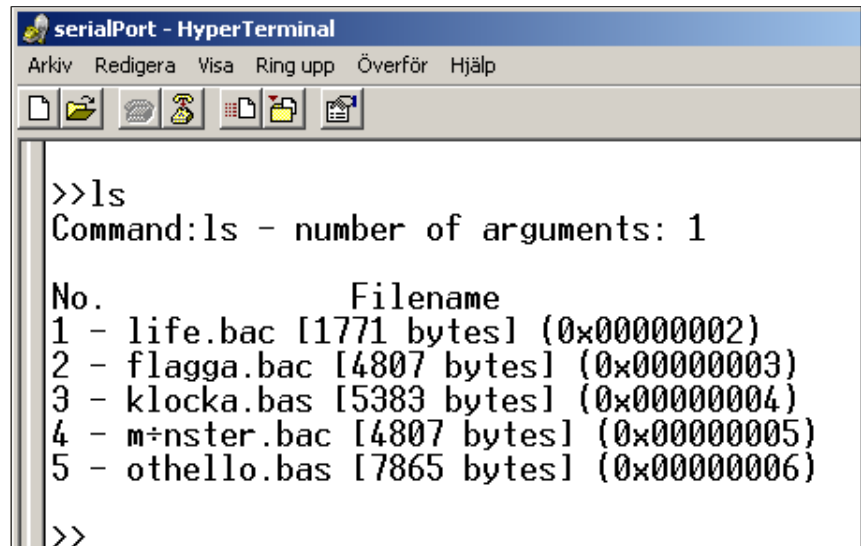


There are however around 12 unused pins of UBW32 (PORTB) that could be used for analog inputs or digital I/O. The abcio.h and abcio.c and disk36.c and myCard.c shows how to emulate an I/O card. Basically, all io strobes from ABC are put in a FIFO that is polled in the main loop. Status (inp1) and a series of data (inp0) can be preloaded. For ABC80 to just read an ADC, one may just define a pointer to a PIC32 register and nothing else needs to be done. It could also be possible to connect a mouse and keyboard to PIC32 and emulate them as ABC80 peripherals. There are some code available from Microchip.

PIC32 debug port

PIC32 debug messages can be sent to the USB virtual com port (optionally TTL port or none, as defined in uarts.h.)

The debug port can also interpret some commands. For instance, one may type **ls** to list the files on SD, or **wifi** to communicate directly with the WiFly module. One may also peek/poke directly into emulated ROM/RAM. Note however that peek/poke into ABC80 internal memory is not possible and that ROM may not be restored by ABC80 reset.



```
>>ls
Command:ls - number of arguments: 1

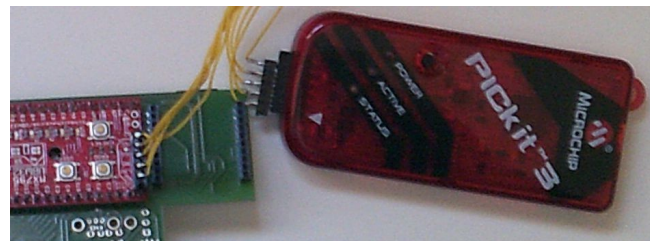
No.      Filename
1 - life.bac [1771 bytes] (0x00000002)
2 - flagga.bac [4807 bytes] (0x00000003)
3 - klocka.bac [5383 bytes] (0x00000004)
4 - m÷nster.bac [4807 bytes] (0x00000005)
5 - othello.bac [7865 bytes] (0x00000006)

>>
```

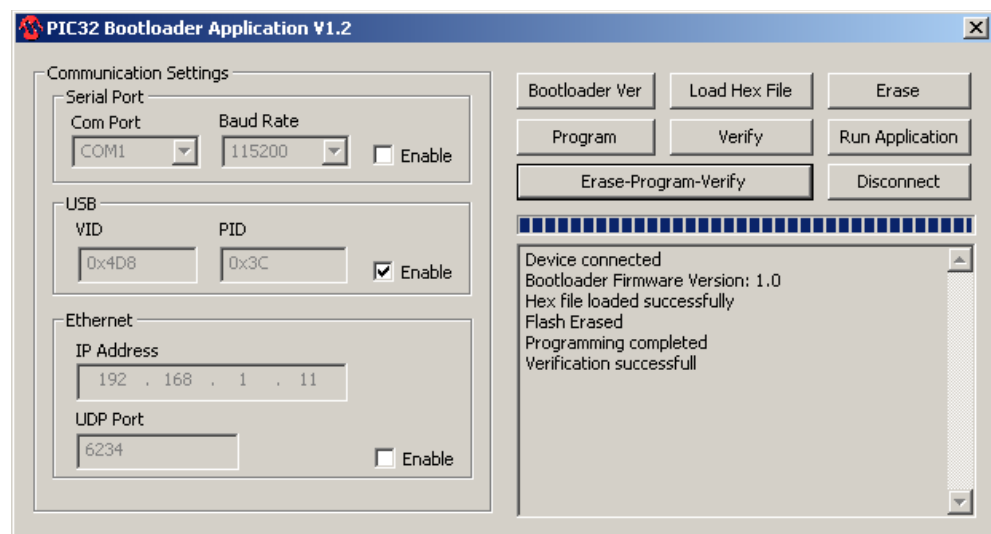
Command parameters must be separated by space, e.g. poke 123 45. The command **iopr 1** (or iopr 2) gives printouts of ABC80 bus io operations as hex code. The format is ioXXYYZZ, where XX=card number, Y=strobe and ZZ=data. The given strobe numbers 0..9 are OUT0..5, Err, INP7, INP0, INP1. For example, if INP0 from card 12 reads 255, this will transmit **io0C8ff**. This is useful for debugging emulated cards and could also be used for testing an external device. Note that the PIC32 memory emulation must be disabled if the external device operates on the memory bus. Disabling the memory emulation can be done by connecting pins RB1-RB0 above the PROG pins.

Firmware updates

A PicKit3 programmer is used for debugging the firmware and for programming the bootloader (straight cable). We use a modified bootloader from Microchip.



Firmware updates can be done via USB while the unit is disconnected from ABC80. Just press the UBW32 prog button while connecting USB (power) and use a special PC software to load the new firmware (hex file).



WiFi

The WiFly RN-XV module and compatible devices can be mounted. Here is an early test by Robert.



Det finns rutiner man kan anropa för att skicka tecken till WiFi och skriva ut inkommande tecken på skärmen. (Detta finns även för TTL/USB, se sd0dev.z)



In this case, Robert is communicating with the WiFly from ABC80.

PIC32 can also route WiFly communication to USB or RS232TTL so that the configuration can be done in a PC terminal program.

Give debug port command **wifi** and PIC32 will route further communication to the WiFly module. Send “\$\$\$” (without CR) and WiFly should respond “CMD”. Note that some programs like TerminalBr@y uses the \$ to define hex code, so one must send \$24\$24\$24 in order to get “\$\$\$”.

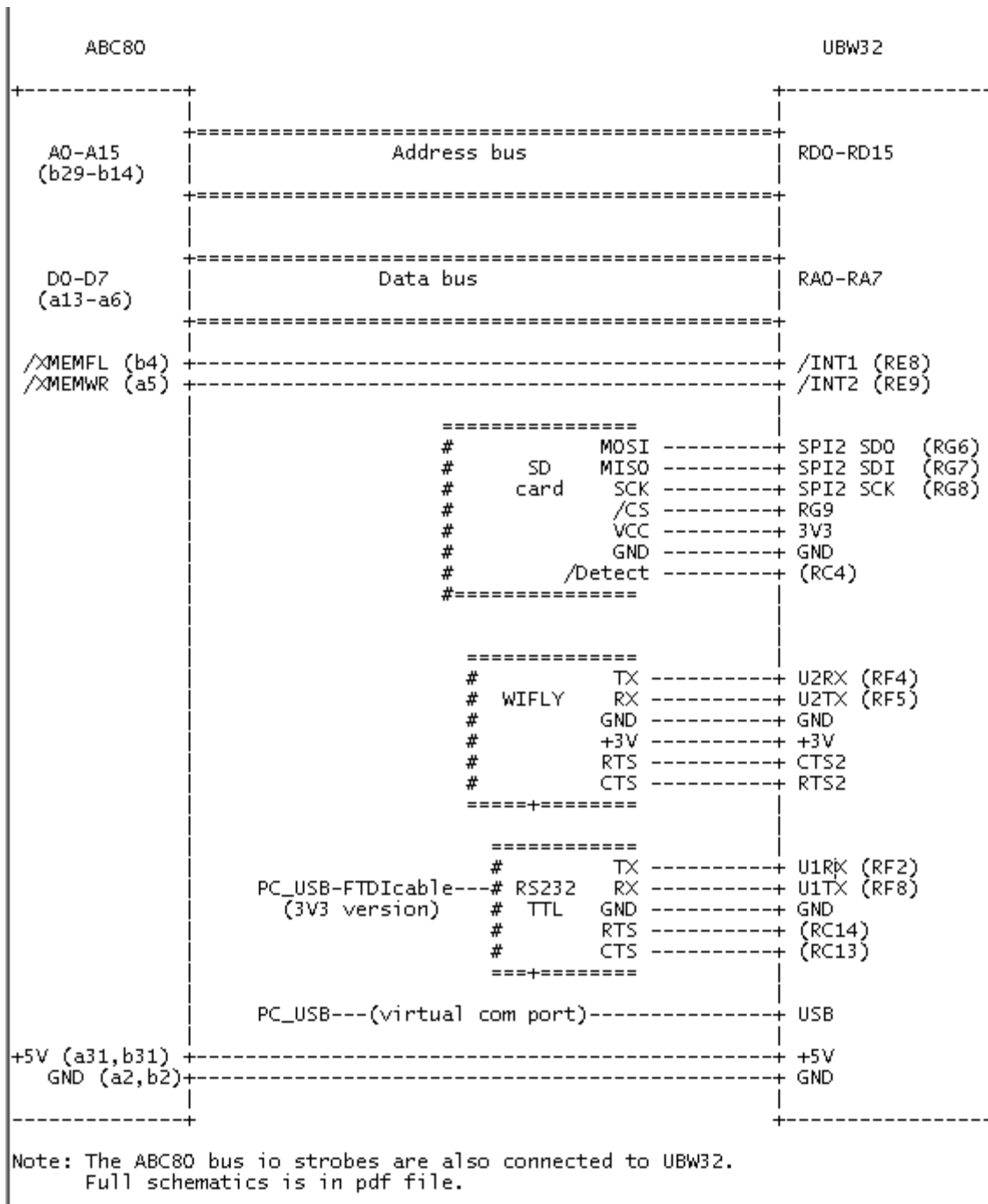
The WiFly should be configured to use RTS and give connect status on GPIO4.

```
set uart flow 1
set sys iofunc 0x10
save
```

```
CLOSE
ERR: ?-Cmd
<2.32> close
<2.32> *CLOSEopen ftp.abc.se 80
Connect to 62.80.200.131:80
<2.32> Connect FAILED
set ftp address 62.80.200.131
ACK
<2.32> set ftp dir /
ACK
<2.32> ftp get README
<2.32> FTP connecting to 62.80.200.131
530 Login incorrect.
FTP ERR
set ftp user anonymous
ACK
<2.32> set ftp pass juhasz@abc.se
ACK
<2.32> ftp get README
<2.32> FTP connecting to 62.80.200.131
*OPEN:FTP timeout=10
```

There are more WiFly settings for IP and ftp. When configuration is done, send '<' to end the WiFly session and return to normal operation to give ABC80 access to the wifly interface.

Schematics



Further info

See README.txt for release notes, copyright and disclaimer.

Email: (Anders) andj@hem.utfors.se, (Robert) juhasz@abc.se, (mailing list) www.abc80.org

Best regards,
Anders & Robert