

Atari USB analog Joystick Driver#

Tested with a Logitech Attack 3 analog Joystick. Other analog Joysticks might work. Please send feedback.

Generic VBI Driver#

Will work with Atari Basic, Turbo Basic, ACTION!, XFORTH.

Compatibility Mode#

Load driver and use Stick(0) and Strig(0) as normal.

Signals from original Joystick in Port A will overwrite USB Joystick compatibility mode.

USB Analog Joystick Mode#

Label	Mem	Values	
USBHORZ	\$0270	horizontal values (127-0 left move, 128 = no move, 129-255 right move)	
USBVERT	\$0271	vertical values (127-0 up move, 128 = no move, 129-255 down move)	
USBTRIG0	\$0272	Buttons 1-8 (each bit, bit set = Button pressed)	
USBTRIG1	\$0273	Buttons 9-10 (each bit, bit set = Button pressed)	
USBTHRUS	\$0274	Thrust (0-255)	

Source Code (BiboAssembler)#

```
01000          .LI OFF
01010 *****
01020 ** 6502 USB DEVELOPMENT **
01030 ** (C) 2004 BY ABBUC **
01040 ** REGIONALGRUPPE FFM **
01050 ** ANALOG JOYSTICK DRIVER **
01060 ** FOR USB SL811HS **
01070 ** VERSION 1.0 20041114 **
01080 ** LICENSED UNDER THE **
01090 ** GNU PUBLIC LICENSE **
01100 ** (GPL) VERS. 2 OR LATER **
01110 ** **
01120 *****
01130 ;
01140          .OR $3500
01150          .OF "D:USBJOYST.COM"
01160 ;
01170 ; SL811 MEMORY ADDRESSES
01180 ; CHANGE ACCORDING TO YOUR
01190 ; CONFIGURATION
01200 USBSEL    = $D500
```

01210 USBDTA = \$D501
01220 ;
01230 ; USB REGISTER SL811
01240 ;
01250 CTL = \$00 ; USBA HOST CTL
01260 BUFADR = \$01 ; BUFFER ADDRESS
01270 BUFLN = \$02 ; BUFFER LEN
01280 PIDEP = \$03 ; HOST PID
01290 PKSTAT = \$03 ; PAKET STATUS
01300 FNADDR = \$04 ; USB ADDR (WO)
01310 MCNTRL = \$05 ; MAIN CONTROL
01320 CDTASET = \$0E
01330 SOFCNT = \$0F ; CNTRL 2 REG
01340 SOFLOW = \$0E ; SOF LOW
01350 INTSTAT = \$0D ; IRQ STATUS
01360 ;
01370 ; USB CONSTANTS
01380 ;
01390 ; INTENA AND INTSTAT MASKS
01400 EP0DONE = \$01
01410 EP1DONE = \$02
01420 EP2DONE = \$04
01430 EP3DONE = \$08
01440 DMADONE = \$10
01450 SOFRECV = \$20
01460 USBRSET = \$40
01470 DMASTAT = \$80
01480 ;
01490 ; ENDPOINT CONTROL REG
01500 EPC0 = \$00 ; ENDPOINT 0
01510 EPC1 = \$10 ; ENDPOINT 1
01520 EPC2 = \$20 ; ENDPOINT 2
01530 EPC3 = \$30 ; ENDPOINT 3
01540 ;
01550 ; ENDPOINT REGISTER OFFSET
01560 ;
01570 EPC = \$00 ; CONTROL
01580 EPBA = \$01 ; BASE ADDRESS
01590 EPBL = \$02 ; BASE LENGTH
01600 EPPS = \$03 ; PACKET STATUS
01610 EPTC = \$04 ; TRANSFERCOUNT
01620 ;
01630 ; PID VALUES
01640 ;
01650 SOFPID = \$05 ; SOF PID
01660 INPID = \$90 ; PACKET ID
01670 SETPID = \$D0 ; SET ADDRESS REQ
01680 ;
01690 ; SET ADDRESS PACKET
01700 ;
01710 SETADDR .HX 0005010000000000
01720 ;
01730 ; SET CONFIG PACKET
01740 ;
01750 SETCONF .HX 0009010000000000
01760 ;
01770 ; ATARI MEMORY LOCATIONS
01780 ;
01790 STICK0 = \$0278

```

01800 STRIG0    = $0284
01810 ;
01820 ; USB JOYSTICK SHADOW REGISTER
01830 ;
01840 USBHORZ  = $0270
01850 USBVERT  = $0271
01860 USBTRIG0 = $0272
01870 USBTRIG1 = $0273
01880 USBTHRUS = $0274
01890 ;
01900 SETVBV   = $E45C
01910 XITVBV   = $E462
01920 VCOUNT   = $D40B
01930 CONSOL   = $D01F
01940 ;
01950 -----
01960 USBRESET
01970         LDA #$AE ; SET SOF
01980         LDX #SOF CNT ; HIGH COUNT
01990         JSR REGSTORE
02000 ;
02010         LDA #$08 ; RESET USB
02020         LDX #MCNTRL ; FULLSPEED
02030         JSR REGSTORE
02040 ;
02050         LDA #$10
02060         JSR PAUSE
02070 ;
02080         LDA #00
02090         LDX #MCNTRL
02100         JSR REGSTORE
02110 ;
02120         RTS
02130 -----
02140 QUERYUSBRESET
02150 ; OUT: A=0 NO USB RESET
02160 ;     A!=0 USBRESET
02170 ;
02180         LDX #INTSTAT
02190         JSR REGFETCH
02200         AND #USBRSET
02210         RTS
02220 -----
02230 CLEARIRQ
02240         LDA #$FF
02250         LDX #INTSTAT
02260         JMP REGSTORE
02270 -----
02280 SPEED
02290 ; OUT: A=0 LOW SPEED DEVICE
02300 ;     A!=0 HIGH SPEED DEVICE
02310 ;         OR ERROR
02320 ;
02330         JSR USBRESET
02340         JSR CLEARIRQ
02350         LDA #10
02360         JSR PAUSE
02370         JSR QUERYUSBRESET
02380         BEQ .1 ; NO RESET

```

```

02390          JSR CLEARIRQ
02400          LDA #$FF
02410          RTS
02420 ;
02430 .1        LDX #INTSTAT
02440          JSR REGFETCH
02450          AND #DMASTAT
02460          BNE .2
02470 ;
02480 ; LOW SPEED
02490 ;
02500          LDA #$AE
02510          LDX #SOF CNT
02520          JSR REGSTORE
02530 ;
02540          LDA #$E0
02550          LDX #CDTASET
02560          JSR REGSTORE
02570 ;
02580          LDA #$05
02590          LDX #MCNTRL
02600          JSR REGSTORE
02610 ;
02620          JSR SETUPUSB
02630          LDA #$00
02640 ;
02650 ; FULL SPEED OR ERROR
02660 ;
02670 .2
02680          RTS
02690 -----
02700 SETUPUSB
02710          LDA #$50
02720          LDX #EPC0+EPPS
02730          JSR REGSTORE
02740 ;
02750          LDA #$00
02760          LDX #EPC0+EPTC
02770          JSR REGSTORE
02780 ;
02790          LDA #$01
02800          LDX #EPC0
02810          JSR REGSTORE
02820 ;
02830          LDA #25
02840          JSR PAUSE
02850 ;
02860          JSR CLEARIRQ
02870          RTS
02880 -----
02890 INITJOYST
02900          LDA #08
02910          LDX #MCNTRL
02920          JSR REGSTORE
02930 ;
02940          LDA #14
02950          JSR PAUSE
02960 ;
02970          LDA #$21

```

```

02980          LDX #MCNTRL
02990          JSR REGSTORE
03000 ;
03010          LDA #$10      ; $10 ADDR
03020          LDX #BUFADR ; DATABUF
03030          JSR REGSTORE
03040 ;
03050          LDA #$8        ; 8 BYTE
03060          LDX #BUFLEN ; DATABUF
03070          JSR REGSTORE
03080 ;
03090          LDA #$E0      ; 1MS EOP
03100          LDX #SOFLOW
03110          JSR REGSTORE
03120 ;
03130          LDA #$EE
03140          LDX #SOFcnt
03150          JSR REGSTORE
03160 ;
03170 ; SET BUFFER FOR SETUP-ADDRESS
03180 ; REQUEST = 1
03190 ;
03200          LDY #8
03210 .1      TYA
03220          CLC
03230          ADC #$F      ; BUF ADDR
03240          TAX
03250          LDA SETADDR-1,Y
03260          JSR REGSTORE
03270          DEY
03280          BNE .1
03290 ;
03300          LDA #00      ; WE USE
03310          LDX #FNADDR ; ADDR 0
03320          JSR REGSTORE
03330 ;
03340          LDA #SETPID
03350          LDX #PIDEP
03360          JSR REGSTORE
03370 ;
03380 .2      LDA #07
03390          JSR PROCESS
03400          AND #04
03410          BNE .2
03420 ;
03430          LDA #20
03440          JSR PAUSE
03450 ;
03460          LDA #INPID
03470          LDX #PIDEP
03480          JSR REGSTORE
03490 ;
03500          LDA #03
03510          JSR PROCESS
03520 ;
03530 ; SELECT CONFIGURATION 1
03540 ;
03550          LDY #8
03560 .3      TYA

```

```

03570      CLC
03580      ADC #$F
03590      TAX
03600      LDA SETCONF-1,Y
03610      JSR REGSTORE
03620      DEY
03630      BNE .3
03640 ;
03650      LDA #01
03660      LDX #FNADDR ; NEW ADDR
03670      JSR REGSTORE
03680 ;
03690      LDA #SETPID
03700      LDX #PIDEP
03710      JSR REGSTORE
03720 ;
03730 .4    LDA #07
03740      JSR PROCESS
03750      AND #04
03760 ;
03770      BNE .4
03780 ;
03790      LDA #INPID
03800      LDX #PIDEP
03810      JSR REGSTORE
03820 ;
03830      LDA #03
03840      JSR PROCESS
03850 ;
03860      LDA #INPID
03870      ORA #01
03880      LDX #PIDEP
03890      JSR REGSTORE
03900 ;
03910      RTS
03920 -----
03930 ; PRINT INLINE STRING
03940 ; END MARKER '@'
03950 ;
03960 PRINT  PLA          get Return address
03970      STA $D0        from Stack
03980      PLA          and store
03990      STA $D1        as pointer
04000 ;
04010 INCP   INC $D0      increase
04020      BNE .1         pointer
04030      INC $D1
04040 .1    LDX #0        read Char from RAM
04050      LDA ($D0,X)
04060      CMP #'@       End?
04070      BEQ ENDPR     yes==>
04080      JSR PUTCHAR   Print Char
04090      JMP INCP     back to loop
04100 ;
04110 ENDPR  LDA $D1      store pointer
04120      PHA          as new
04130      LDA $D0      return address
04140      PHA          on stack
04150      RTS          continue pgm

```

```

04160 ;                after text
04170 -----
04180 PUTCHAR  TAX          Print char
04190          LDA $E407    with OS
04200          PHA          Routine
04210          LDA $E406
04220          PHA
04230          TXA
04240          RTS          JUMP
04250 -----
04260 WAITJOYSTICK
04270          JSR PRINT
04280          .HX 9B
04290          .AS "ATARI USB JOYSTICK DRIVER"
04300          .HX 9B
04310          .AS "Version 1.0 / GNU License"
04320          .HX 9B
04330          .AS "(c) 2004 ABBUC e.V."
04340          .HX 9B
04350          .AS "H. Reminder, T. Grasel, C. Strotmann"
04360          .HX 9B9B
04370          .AS "WAIT FOR DEVICE, [START] TO SKIP..."
04380          .HX 9B40
04390 .1          JSR SPEED
04400          BEQ .2
04410 ; QUERY  CONSOL KEYS
04420          LDA CONSOL
04430          AND #1 ; START KEY
04440          BEQ .3 ; SKIP USB
04450          BNE .1
04460 ;
04470 .2          JSR PRINT
04480          .AS "LOW SPEED DEVICE DETECTED!"
04490          .HX 9B40
04500 ;
04510          JSR INITJOYST
04520          JSR PRINT
04530          .AS "JOYSTICK INITIALIZED."
04540          .HX 9B40
04550          LDX /JOYVBI
04560          LDY #JOYVBI
04570          LDA #7
04580          JSR SETVBV
04590          RTS
04600 ;
04610 .3          JSR PRINT
04620          .AS "USB detection skipped,"
04630          .HX 9B
04640          .AS "no USB Driver installed!"
04650          .HX 9B40
04660          RTS
04670 -----
04680 RESPART  .OR $600
04690 ;
04700 ; THRESHOLD ANALOG->DIGITAL
04710 THLEFT   .HX 60
04720 THRIGHT  .HX A0
04730 THUP     .HX 60
04740 THDOWN   .HX A0

```

```

04750 -----
04760 REGFETCH
04770 ; IN:  X=USB REGISTER
04780 ; OUT: A=USB DATA
04790         STX USBSEL
04800         LDA USBDTA
04810         RTS
04820 -----
04830 REGSTORE
04840 ; IN:  A=USB DATA
04850 ;      X=USB REGISTER
04860         STX USBSEL
04870         STA USBDTA
04880         RTS
04890 -----
04900 PAUSE
04910 ; IN:  A=NUMBER OF 1/50 SEC
04920         TAX
04930 .1      LDA VCOUNT
04940         BNE .1
04950         DEX
04960         BNE .1
04970         RTS
04980 -----
04990 GETJOYSTICK
05000 ;
05010         LDA #03
05020         JSR PROCESS
05030 ;      AND #01
05040 ;      BEQ .2 ; NO DATA
05050 ;
05060         LDX #$10
05070         JSR REGFETCH
05080         STA USBHORZ
05090         LDX #$11
05100         JSR REGFETCH
05110         STA USBVERT
05120         LDX #$12
05130         JSR REGFETCH
05140         STA USBTHRUS
05150         LDX #$13
05160         JSR REGFETCH
05170         STA USBTRIG0
05180         LDX #$14
05190         JSR REGFETCH
05200         STA USBTRIG1
05210 ;
05220 .2      RTS
05230 -----
05240 PROCESS
05250 ; IN:  A=USB COMMAND
05260 ; OUT: A=RETURNCODE
05270         PHA
05280         LDA #01
05290         LDX #INTSTAT
05300         JSR REGSTORE
05310 ;
05320         PLA
05330         LDX #CTL

```



```

05340          JSR REGSTORE
05350 ;
05360 .1        LDX #INTSTAT
05370          JSR REGFETCH
05380          AND #$01
05390          BEQ .1
05400 ;
05410          LDX #PKSTAT
05420          JSR REGFETCH
05430          RTS
05440 -----
05450 USB2ATA
05460          LDA #$0F
05470          STA STICK0
05480          LDA #1
05490          STA STRIG0
05500 ;
05510          LDA USBTRIG0
05520          ORA USBTRIG1
05530          BEQ GETSTICK
05540          LDA #0
05550          STA STRIG0
05560 ;
05570 GETSTICK
05580          LDA STICK0
05590          LDX USBHORZ
05600          CPX THLEFT
05610          BCC .1 ; LEFT
05620          CPX THRIGHT
05630          BCS .2 ; RIGHT
05640          BCC .10
05650 .1        EOR #$04 ; LEFT
05660          BPL .3
05670 .2        EOR #$08 ; RIGHT
05680 .3        STA STICK0
05690 .10
05700          LDX USBVERT
05710          CPX THUP
05720          BCC .11 ; UP
05730          CPX THDOWN
05740          BCS .12 ; DOWN
05750          BCC .20
05760 .11       EOR #$01 ; UP
05770          BNE .13
05780 .12       EOR #$02 ; DOWN
05790 .13       STA STICK0
05800 .20       RTS
05810 -----
05820 JOYVBI
05830          LDA STICK0
05840          EOR #$0F
05850          BNE .1
05860          LDA STRIG0
05870          BEQ .1
05880 ;
05890          JSR GETJOYSTICK
05900          JSR USB2ATA
05910 .1        JMP XITVBV
05920 -----

```

05930 .OR \$2E0
05940 .DA WAITJOYSTICK
05950 -----