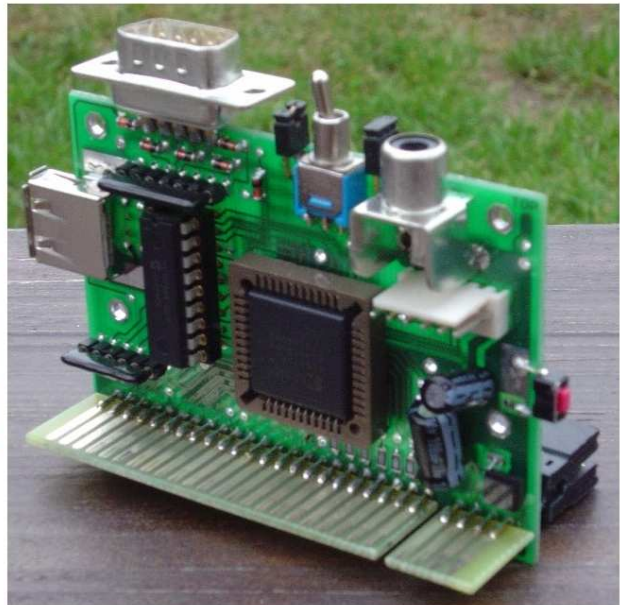
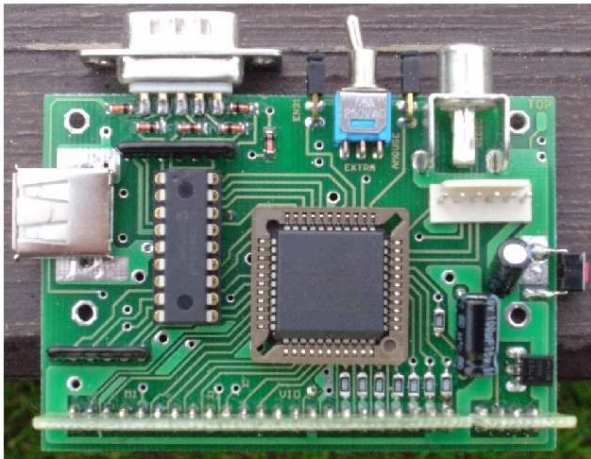
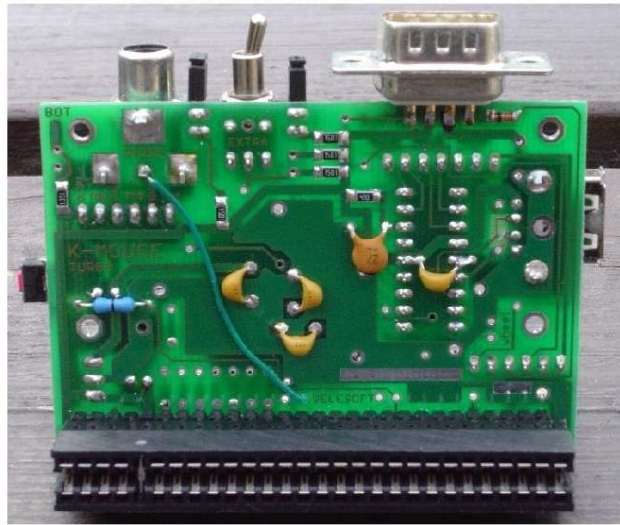
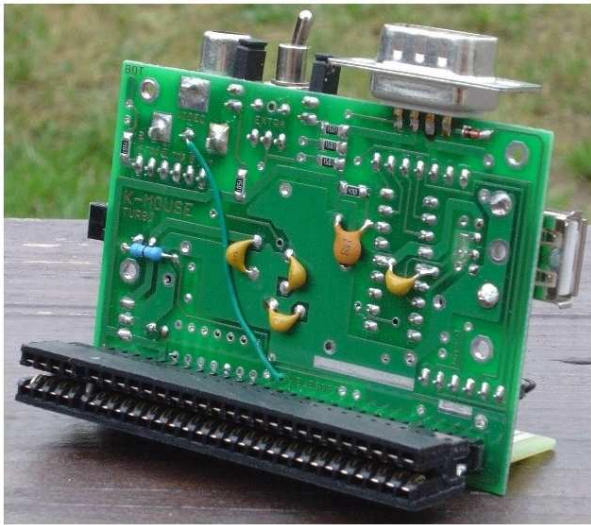


verze 2005-3 (23.11.2005)



KEMPSTON MOUSE TURBO INTERFACE



KEMPSTON MOUSE TURBO je kompatibilní s originál Kempston mouse interface. Narozdíl od originálu má několik dalších vylepšení, ale je zpětně plně kompatibilní. Rusové ve svém zapojení Kempston mouse, které okopírovali z originálu, podporují již tří tlačítkovou myš (originál měl jen 2 tlačítka). Náš "turbo" interface umožňuje připojit PS/2 (nebo USB s protokolem PS/2) myš. Podporuje až 4 tlačítka a dokonce i kolečko. Nyní si povíme něco o tom, jak vlastně Kempston mouse interface funguje a na jakých portech je čitelný. O komunikaci s myší se stará samotný interface, uživatel jen čte z patřičných portů aktuální souřadnice pro osu X a Y, nebo stav tlačítek. Každá osa má rozsah od 0 do 255(1 bajt). Po prvním zapnutí interface jsou souřadnice v obou osách 0. Při pohybu myši vpravo je hodnota osy X postupně inkrementována, při pohybu vlevo se hodnota naopak snižuje. To samé platí i pro osu Y, která je inkrementována při pohybu dolů. Teď se podíváme například na hodnoty osy X při delším pohybu myši vpravo. Postupně se zvyšuje souřadnice z 0, 1, 2, 3 ... až do 255, což je vlastně maximální hodnota jednoho bajtu. Interface ale neošetřuje mezní hodnoty, takže se při pokračování pohybu vpravo souřadnice(bajt) dál zvyšuje a po 255 "přeteče" znovu do 0 a pokračuje dál (1, 2, 3 ...255). To platí i při pohybu vlevo, kdy se hodnota snižuje až do 0 a pak

"přeteče" do 255 a dál se snižuje (254, 253, 252 ...0). Z portu 64223 zjistíme aktuální stav všech tlačítek (stisknuté / povolené) ale to už nám přiblíží následující přehled.

OSA X je čitelná na portu 64479 (#FBDF)

OSA Y je čitelná na portu 65503 (#FFDF)

TLAČÍTKA jsou čitelná na portu 64223 (#FADF)

Čtená data z portu:

D0 = pravé tlačítko

D1 = levé tlačítko

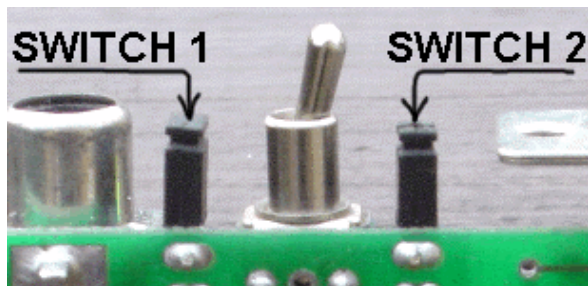
D2 = prostřední tlačítko (používá jen RUSKÁ a TURBO K-MOUSE)

D3 = 4.tlačítko (používá jen TURBO K-MOUSE)

D4 až D7 = 4-bitový binární čítač polohy kolečka

Jednotlivé bity představují tlačítka myši. Přitom bit v log.0 značí stisknuté a log.1 nestisknuté. Jak už jsme si řekli, původní Kempston mouse interface používal jen dvě tlačítka (D0 a D1). Ruská Kempston mouse používá tlačítka tři (D0, D1 a D2). K-MOUSE TURBO používá celkem 4 tlačítka (D0, D1, D2, D3). Na D4 - D7 je čitelná 4-bitová hodnota polohy kolečka. Tento čítač pracuje stejně jako osy X a Y. Při pohybu kolečka směrem dolů se čítač inkrementuje (0, 1, 2, ...,14, 15, 0, 1). Tak to by bylo k hlavním portům myši.

AMIGA MOUSE a KEMPSTON JOYSTICK na portu 31



| SWITCH 1 | SWITCH 2 | FUNKCE PORTU 31 |
|----------|----------|------------------------|
| ? | OFF | port není čitelný |
| OFF | ON | port vrací hodnotu 0 |
| ON | ON | amiga mouse / joystick |

Interface obsahuje jeden přepínač, dva switche, konektor CANON 9, tlačítko reset a konektor cinch (VIDEO). Pravý switch (SWITCH 2) slouží k povolení portu 31, na kterém může být čitelná buď amiga mouse / kempston joystick, nebo hodnota 0. Pokud je SWITCH 2 odstraněn, port 31 není přístupný. Levý switch (SWITCH 1) určuje, jestli na portu 31 bude emulována amiga mouse / joy (switch zapojen), nebo bude nulován (switch nezapojen). Nulování portu 31 zabrání některým hrám v detekci fire joysticku. Mezi amiga mouse a kempston joystickem se přepíná jednoduše stiskem tlačítka myši (bude aktivní amiga mouse) nebo fire na joysticku

(bude aktivní joystick) zapojeném do konektoru CANON 9. Při emulaci amiga mouse mají data portu 31 následující uspořádání:

AMIGA MOUSE:

D0 = XB

D1 = YB

D2 = XA

D3 = YA

D4 = levé tlačítko

D5 = pravé tlačítko

D6 = prostřední tlačítko

D7 = čtvrté tlačítko (jen K-MOUSE TURBO)

Na D0 a D2 jsou přímo stavy clonek myši pro osu X, na D1 a D3 pro osu Y. Na původně nevyužitém bitu D7 je nově přidané čtvrté tlačítko amiga myši. Při stisknutém tlačítku je bit v log.1. Stiskem fire joysticku se odpojí amiga myš a bude čitelný pouze kempston joystick. Pak platí klasické rozložení :

KEMPSTON JOYSTICK:

D0 = joy VPRAVO

D1 = joy VLEVO

D2 = joy DOLU

D3 = joy NAHORU

D4 = joy FIRE 1

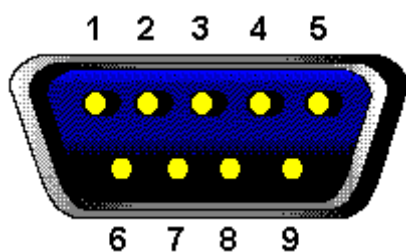
D5 = joy FIRE 2

D6 = joy FIRE 3

D7 = vždy log.0

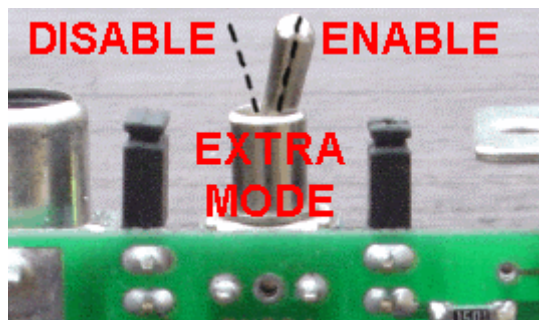
Při aktivním směru nebo fire je odpovídající bit nastaven na log.1. Po aktivaci joysticku přestává pracovat i K-MOUSE/A-MOUSE. Z čehož plyne, že není možné současně používat myš i joystick. K-MOUSE IF má datovou sběrnici kompletně oddělenou odpory, což zabraňuje možným kolizím s jinými perifériemi. Interface je tedy možné provozovat s jakoukoli jinou periférií. Pokud přece jen připojena periferie "koliduje" s porty K-MOUSE IF, vnutí jen svoje data, a porty K-MOUSE IF nebudou čitelné. Například u D40/D80 bude na portu 31 čitelná 8255 (pokud je povolena) a port 31 u K-MOUSE IF nebude přístupný. Při softwarovém zablokování 8255 (v D40/D80) bude na portu 31 zase K-MOUSE IF. Na obrázku je popis zapojení konektoru CANON 9 pro kempston joystick. Konektor CANON je plně zapojený, takže funguje i třítlačítkový joystick s autofire.

KONEKTOR CANON



| | |
|----|--------|
| 1- | UP |
| 2- | DOWN |
| 3- | LEFT |
| 4- | RIGHT |
| 5- | FIRE 3 |
| 6- | FIRE 1 |
| 7- | +5V |
| 8- | GND |
| 9- | FIRE 2 |

EXTRA MÓD



Extra mód je vymoženost K-MOUSE turbo interface. Slouží k nastavení vlastností myši. Dvoupolohový přepínač je určený k povolení nebo blokování extra

módu. Při blokování není možné dále vyvolat extra mód a tedy ani měnit vlastnosti interface. Při povolení (svítí žlutá LED dioda na přepínače) lze kdykoli za provozu stiskem a povolením levého + pravého + prostředního tlačítka vyvolat extra mód. V extra módu následným stiskem jednoho z tlačítek změníte vlastnosti interface. Tlačítka mají tento význam:

Levé tlačítko = nuluje port 31, nebo na portu 31 nechá stav podle naswitchování

Pravé tlačítko = prohodí funkci levého a pravého tlačítka (jen u K-MOUSE)

Prostřední tlačítko = sníží /zvýší 4x rychlost pohybu myši (jen u K-MOUSE)

Po uvolnění stisknutého tlačítka v extra módu se automaticky extra mód opustí. Nastavení všech vlastností je možné resetovat opětovným stiskem a uvolněním všech třech tlačítek v extra módu. V době, kdy je extra mód aktivní, nejsou na portu K-MOUSE (64223) čitelná tlačítka, současně je na portu 31 čitelná jen hodnota 0. Tato vlastnost znemožní softwarovou detekci stisku tlačítek myši, pokud je uživatel v extra módu. Majitele gumáků jistě potěší možnost osazení interface tlačítkem reset, nebo konektorem cinch s videovýstupem. Videovýstup funguje jen na ZX16 / ZX48 / ZX48+, u ostatních modelů není signál video vyveden na sběrnici ZX Spectra.

Dodatečně lze osadit ještě jeden switch vedle obvodu PIC (vedle nápisu WHEEL). Zapojením switche se vypne funkce kolečka a na bitech D4-D7 portu tlačítek K-MOUSE bude čitelná log.1.

VIDEO VÝSTUP

Interface lze osadit konektorem CINCH pro video výstup. Bohužel signál video mají vyvedený na sběrnici pouze ZX Spectra 16kB a 48kB. Ostatní 128kB modely mají stejný pin sběrnice nezapojen. Prakticky u všech verzí ZX Spectra 16/48kB je na sběrnici značně zarušená zem. Ze sběrnice tedy kvalitní video nelze vyvést. Problém se z 80% vyřeší, pokud propojíte kablíkem GND sběrnice se zemí kdekoli poblíž TV modulátoru(nebo přímotělo modulátoru). Obraz se stabilizuje a pruhy nejsou tak rušivé. Některé starší verze ZX, zejména ISSUE TWO, nemají signál video na sběrnici přiveden. Stačí zde zapájet drátovou propojkou označenou „VID“ umístěnou mezi ulou a LM1889.

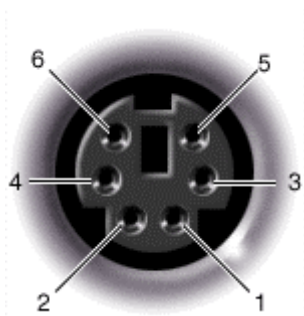
ADRESACE PORTŮ

KEMPSTON MOUSE TURBO interface používá všechny své porty úplně adresované.

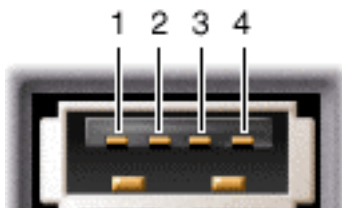
| <u>Funkce</u> | <u>port</u> | <u>adresace</u> | <u>portu binárně (od A15 do A0)</u> |
|---------------|---------------|-----------------|-------------------------------------|
| tlačítka | 64223 (#FADF) | xxxxx0x0 | 11011111 |
| osa X | 64479 (#FBDF) | xxxxx0x1 | 11011111 |
| osa Y | 65503 (#FFDF) | xxxxx1x1 | 11011111 |
| a-mouse/joy | 31 (#1F) | xxxxxxxx | 00011111 |

PS/2 NEBO USB ?

Na desku interface můžete osadit buď PS/2, nebo USB konektor. Po osazení je nutné na spodní straně plošného spoje přerušit některé vodiče vedoucí na piny připájeného konektoru. V každém případě se tento zákrok musí provést ještě před prvním zapnutím interface, jinak dojde ke zkratu na pinech obvodu PIC ! Pro úplnost uvádím opět pinout obou konektorů.



- 1 - data
- 2 - nezapojeno
- 3 - GND
- 4 - +5V
- 5 - clock
- 6 - nezapojeno



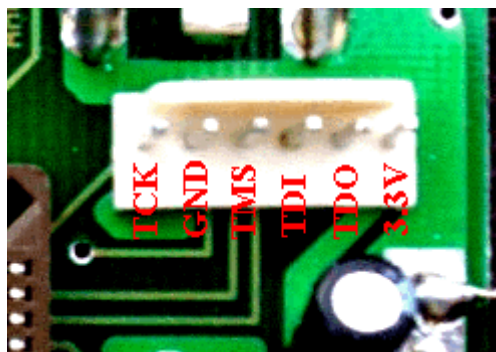
- 1 - +5V
- 2 - data
- 3 - clock
- 4 - GND

EXTERNÍ A INTERNÍ VERZE

Externí verze má už přímo na desce napájené všechny konektory. Hlavní obvody XILINX XC9572XL a PIC jsou osazeny v patičkách. Při poruše stačí obvod jednoduše vyměnit za nový. Není nutný zásah do počítače. Může se kdykoli odpojit a použít u jiného počítače. U počítačů Didaktik M a Didaktik Kompakt je vystouplý profil nad sběrnici, a bude třeba mezi interface a konektor zapájet krátký mezikus který sběrnici prodlouží. Interní verze se zatím připravuje. Je určena pro zapojení přímo na desku počítače. Musíme si nejprve najít vhodné místo pro umístění,

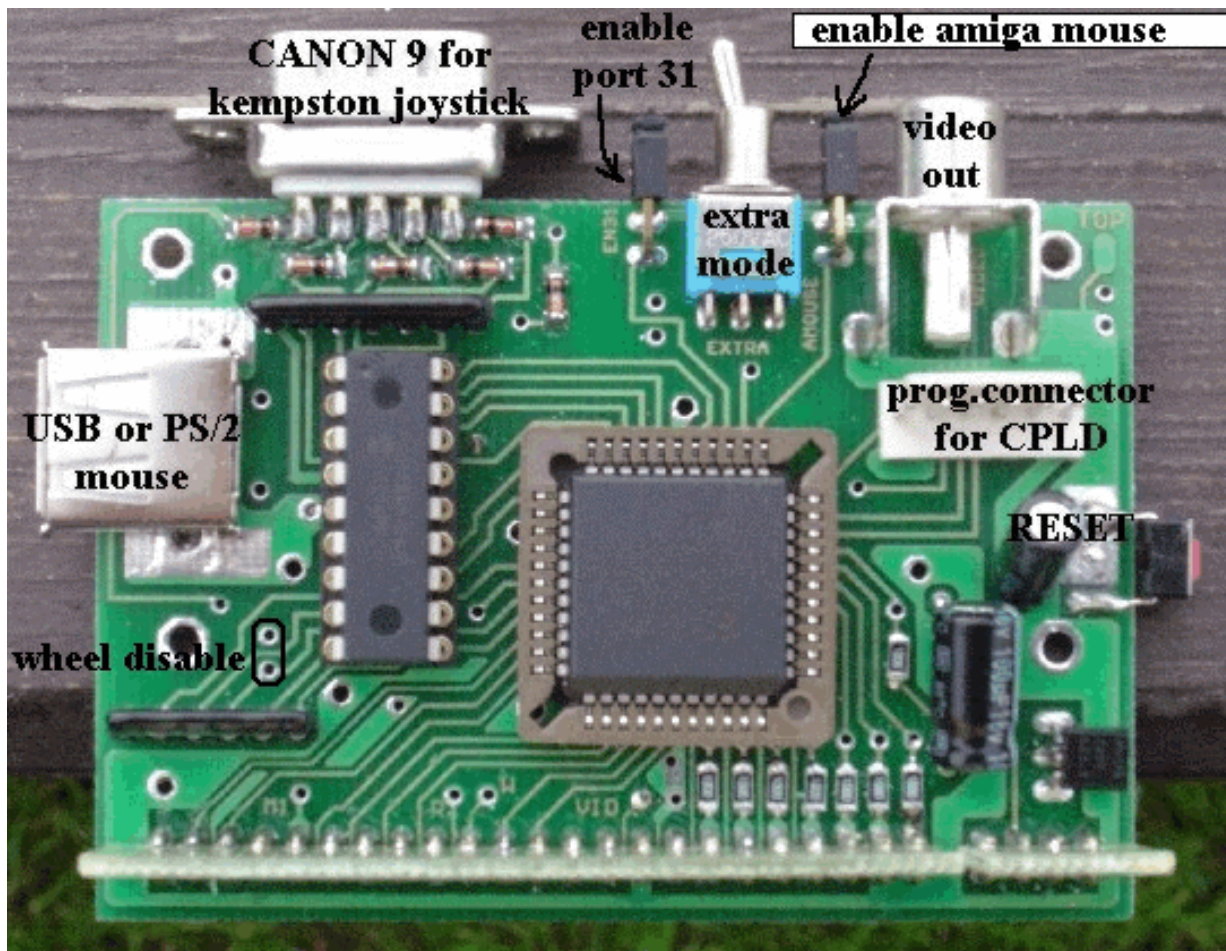
tak aby šel zavřít horní kryt. Tato verze bude kompletně v SMD provedení na malé desce. Interface se připojí pomocí plochého výčežilového kabelu k signálům procesoru, ostatní přepínače a konektory se propojí rovněž plochým kabelem k desce interface. Plochý kabel k procesoru musí být pokud možno co nejkratší! Interní verze určitě půjde připojit do Didaktiků M/Kompakt, ZX Spectrum+ a ZX128+2, ve kterých je dostatek místa. Na druhou stranu například do klasického ZX Spectra (GUMÁK) se už asi nic interního nevejde. Nevýhodou je nutnost pájet v počítači hromadu drátů. SMD součástky budou připájené přímo na desce a budou se obtížně měnit (při případné poruše).

PŘEPROGRAMOVÁNÍ INTERFACE



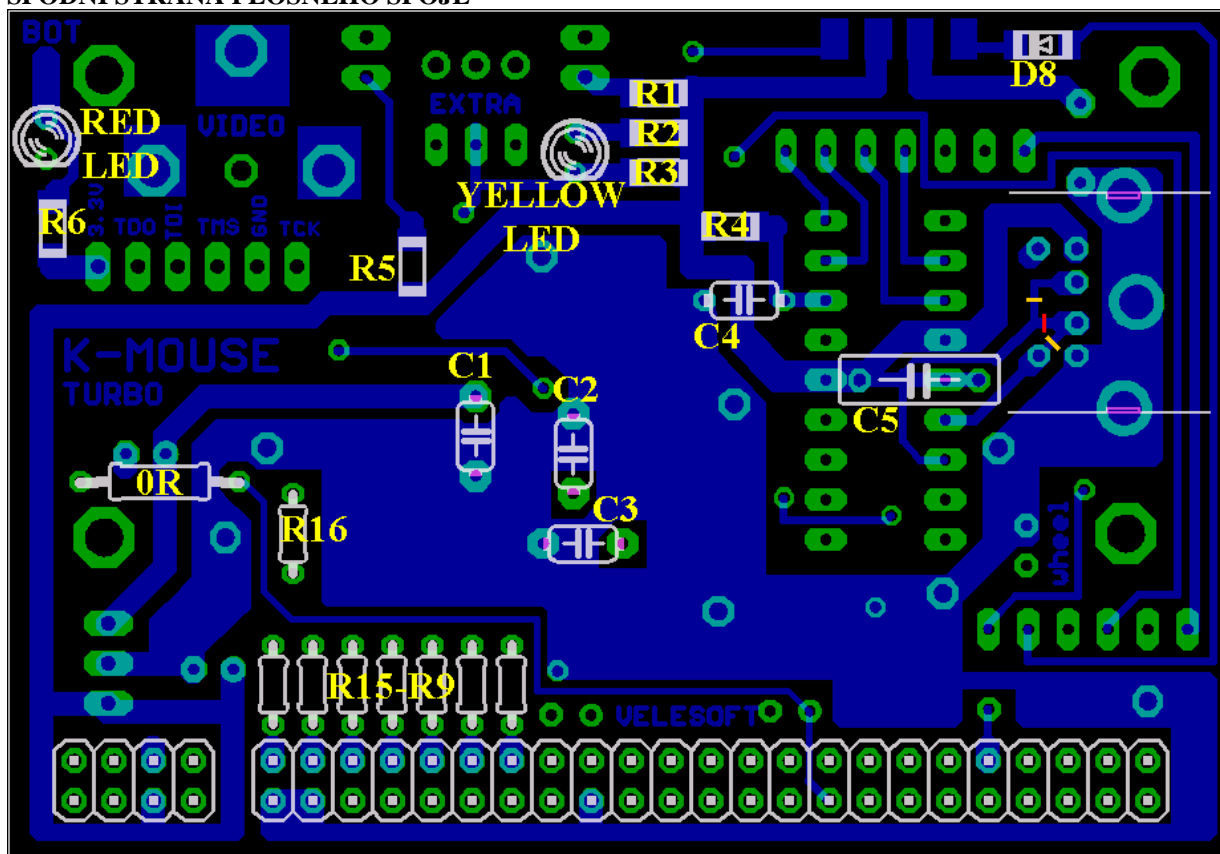
Deska obsahuje konektor (6 pinů) pro připojení k programátoru obvodů XILINX CPLD. Konektor je zapojený podle kabelu vývojového kitu KRAKEN, který si můžete objednat na WWW.MCU.CZ Signály konektoru jsou u jednotlivých pinů na desce popsány. Obsah JEDEC je přiložen k dokumentaci.

OSAZENÍ PLOŠNÉHO SPOJE



Na desku osadíme patici DIL 18, určenou pro obvod PIC. Ten můžeme být typu PIC 16F84A nebo PIC16F627 / PIC16F628, a to v pouzdrú PDIP18 ve verzi 4MHz. Při použití PIC16F627 / 16F628 neosazujeme rezistor R4 a kondenzátor C4, nebo jen jednoduše u PICu nožku č.16 nezasuneme do patice. Konektor pro sběrnici ZX Spectra použijeme SLOT 62, zkrácený o 4 krajní piny. Konektor cinch a tlačítko reset není nutné osazovat. Pro obvod CPLD osadíme patici PCCL44Z (pinem 1 dolů).

SPODNÍ STRANA PLOŠNÉHO SPOJE



Seznam součástek pro osazení spodní strany plošného spoje:

RED LED – červená led dioda 3mm/2mA(L-3MM2MA/R). Pokud svítí, je stabilizátor LP2950CZ v pořádku.

YELLOW LED – žlutá led dioda 3mm/2mA(L-3MM2MA/Y). Signalizuje povolený extra mód.

R1-R3,R5,R6 – odpor 1K5 v provedení SMD (velikost 1206)

R4 - odpor 4K7 Ohmů v provedení SMD (velikost 1206)

R9-R16 – odpor 150 Ohmů. Chrání datovou sběrnici CPLD před kolizí s jinou periferií. Lze použít buď klasické odpory(velikostí vyhovují „metalizované miniaturní rezistory“ o délce těla 3.5mm) osazené na této spodní straně plošňáku, nebo odpory SMD(opět velikost 1206) osazené na horní straně plošného spoje.

C1-C3 – keramický kondenzátor 47nF(osadíme až po připájení patice PCCL, jinak mohou vadit dlouhé vývody)

C4 – keramický kondenzátor 27pF(C4 a R4 generuje hodinový signál 4MHz pro PIC16F84)

C5 – keramický kondenzátor 100nF

D8 – dioda 1N4148 v provedení SMD(velikost 1206). Pokud nebudete osazovat konektor canon, není třeba.

0R – odpor 0 Ohmů můžeme nahradit třeba drátovou propojkou. Přivádí signál reset na pájecí plošky pro tlačítko.

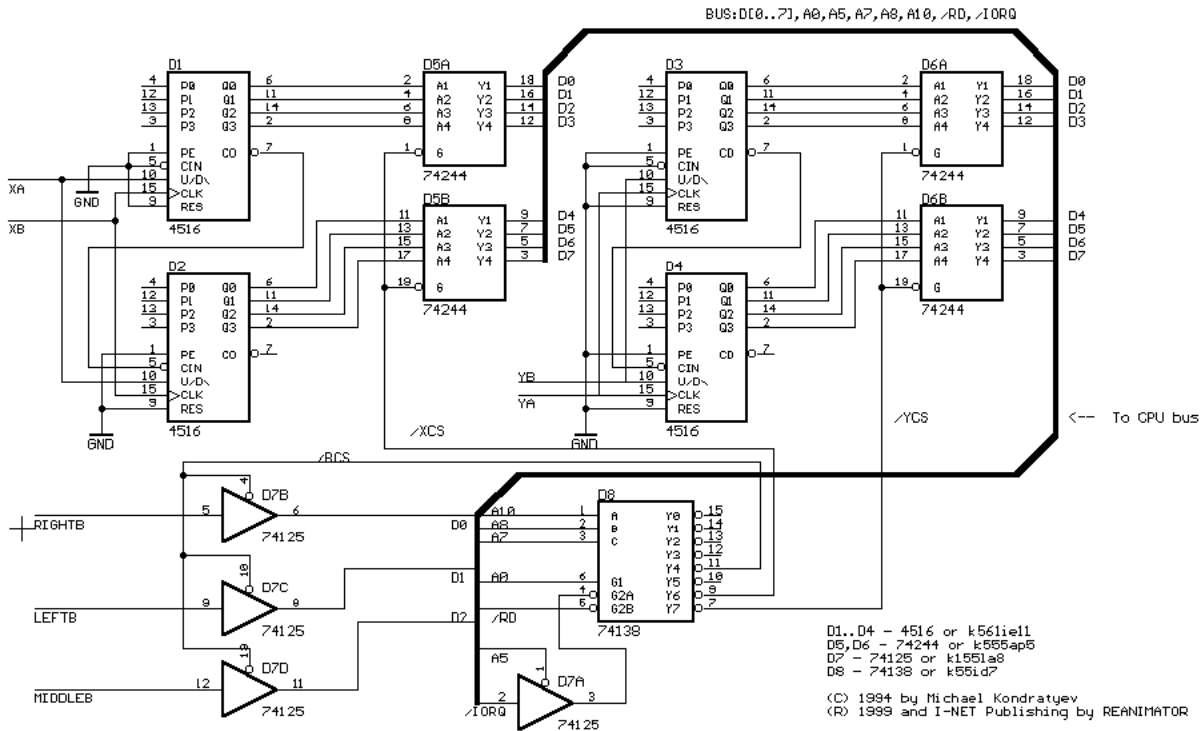
U konektoru USB/PS2 jsou barevně přeškrtnuté vodiče. Při osazení USB konektoru přerušíme spoj v bodě červené značky, při osazení PS/2 konektoru přerušíme dva spoje v místě značek oranžových.

Ze spodní strany spoje nakonec osadíme zkrácenou dvouřadou pinovou lištu **S2G80** delším vývodem skrz prokované otvory a takto ji připájíme z horní strany desky. Nyní osadíme na vrchní straně desky průchozí sběrnici tak, aby mezera pro klíč byla dostatečná. Nejlepší je zasunout oba díly průchozí sběrnice do slotu a desky přisunout až na doraz ke klíči. Takto zasunutě desky necháme ve slotu a připájíme na pinovou lištu z vrchní strany. Na lištu spodní strany desky připájíme **SLOT** s přihnutými vývody k sobě, tak, aby šel dobře pájet. Pozor, aby souhlasila strana s klíčem ! Všechny diody a konektor **CANON** (CAN9V) se osazují jen kvůli joysticku, jinak nejsou povinné. Piny 1-5,6 a 9 konektoru canon je třeba přes odpory kolem 4K7 spojit s +5V (CANON pin č.7). Nejvhodnější je použít dvě odporové sítě se společným vodičem. Tím docílíme pull-upu všech signálů joysticku a na deším kabelu joye se už nebudou zakmitávat směry. CPLD je na zákmity dost citlivé a i když joystick nebude aktivován, k zákmitům docházet může. Konektor canon je paralelně spojený s vývody obvodu PIC, který na stejných vodičích posílá do CPLD stavy slonek při práci s myší. Každý zákmit signálu z joysticku bere CPLD jako změnu směru pohybu myši, což má za následek, že se mění souřadnice v osách K-MOUSE.

POPIS FUNKCE

O komunikaci s PS/2 myši se stará programovatelný obvod PIC16F84, případně 16F627 nebo 16F628, jehož autorem je Denis Sotchenko (rdc@cyberlan.mtu-net.ru). PIC pracuje jako konvertor z PS/2 na AMIGA mouse. Bližší informace o tomto konvertoru můžete najít třeba na: <http://ps2m.sotchenko.ru/index.html> nebo stáhnout přímo z Aminetu : <ftp://ftp.wustl.edu/pub/aminet/hack/hack/ps2m.lha> . A pro úplnost ještě odkaz na soubor HEX s obsahem pro PIC: http://ps2m.sotchenko.ru/pic/ps2m_5button.hex . K-MOUSE interface obsahuje ještě druhý důležitý obvod CPLD (XILINX XC9572XL) jehož autorem jsem já. Moje CPLD se stará o konverzi signálů AMIGA myši na KEMPSTON MOUSE. Tento obvod umí pracovat i jako transparentní kvůli možnosti číst na portu 31 stavy clonek AMIGA myši. Po osazení PICu můžeme používat běžnou PS/2 myš a u K-MOUSE tak podporovat i čtvrté tlačítko + kolečko myši. Při návrhu obsahu CPLD jsem vycházel z původního zapojení ruské KEMPSTON MOUSE, které si můžete prohlédnout na následujícím obrázku.

SCHEMA RUSKÉ KEMPSTON MOUSE



Takové zapojení je sice jednoduché, levné a funkční, ale souřadnice pro Kempston mouse mění 4x pomaleji než by mohlo. Zapojení jsem vylepšil a myš pracuje 4x rychleji (stejně jako u většiny emulátorů ZX). Některé programy jsou ovšem dost náchylné na rychlejší pohyby myši. To je přesně příklad ARTISTu 2. Ovladač pro K-MOUSE tam zpracovává jen malé odchylky při pohybu, a při větší rychlosti začne šipka dost nesmyslně poskakovat. Proto jsem do K-MOUSE TURBO interface přidal možnost přepnutí citlivosti myši podle původního zapojení. Originál Kempston mouse byla stejně pomalá, jako ruská kopie.

INFORMACE O ORIGINÁL KEMPSTON MOUSE INTERFACE

Před nějakým časem jsem měl možnost testovat originál Kempston mouse interface. Při té příležitosti jsem zmapoval schema zapojení a otestoval skutečnou adresaci portů. Reálné testy adresace portů se neshodují s žádným materiálem z internetu.

Dokonce i na www.worldofspectrum.org je adresace chybná:

Funkce **port** **adresace portu binárně (od A15 do A0)**

osa X 64479(#FBDF) xxxxx0x1 xx0xxxxx

osa Y 65503(#FFDF) xxxxx1x1 xx0xxxxx

tlačítka 64223(#FADF) xxxxxxx0 xx0xxxxx

Skutečná adresace portů původní Kempston mouse je tedy:

| Funkce | port | adresace portu binárně (od A15 do A0) |
|----------|--------------|---------------------------------------|
| osa X | 64479(#FBDF) | xxxxx011 xx0xxxxx |
| osa Y | 65503(#FFDF) | xxxxx111 xx0xxxxx |
| tlačítka | 64223(#FADF) | xxxxxx10 xx0xxxxx |

Připojená myš vrací na konektoru canon stavy cloněk pro osu X,Y a stav tlačítek. Funkčně je shodná s amiga mouse, má pouze prohozené signály na konektoru.

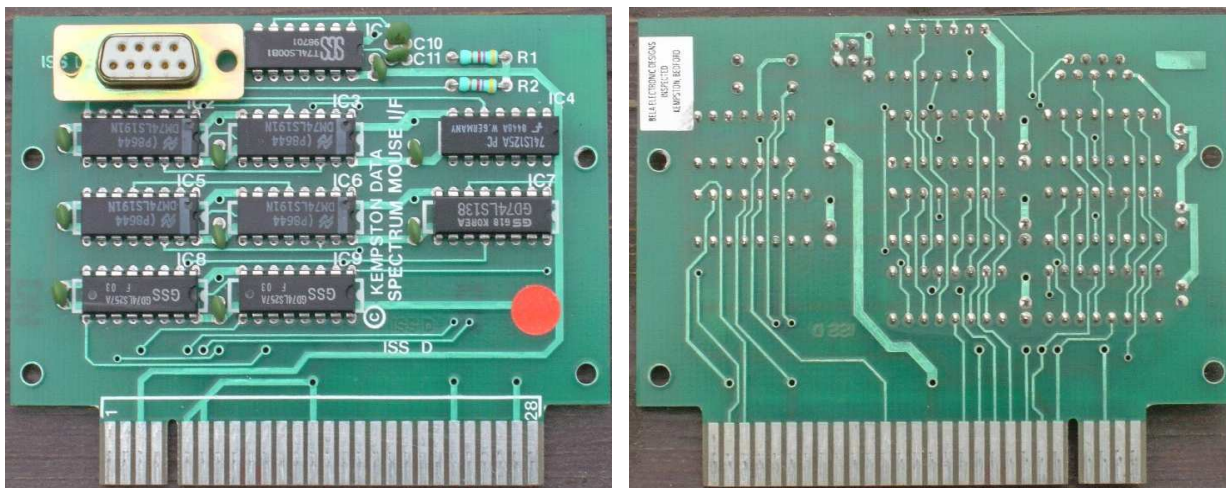
Pinout konektoru canon původní kempston myši:

| PIN | SIGNÁL |
|-----|-------------------------------|
| 1 | +5V |
| 2 | clonka osy X |
| 3 | clonka osy Y |
| 4 | levé tlačítko(log.0 = stisk) |
| 5 | pravé tlačítko(log.0 = stisk) |
| 6 | GND |
| 7 | nezapojeno |
| 8 | clonka osy Y |
| 9 | clonka osy X |

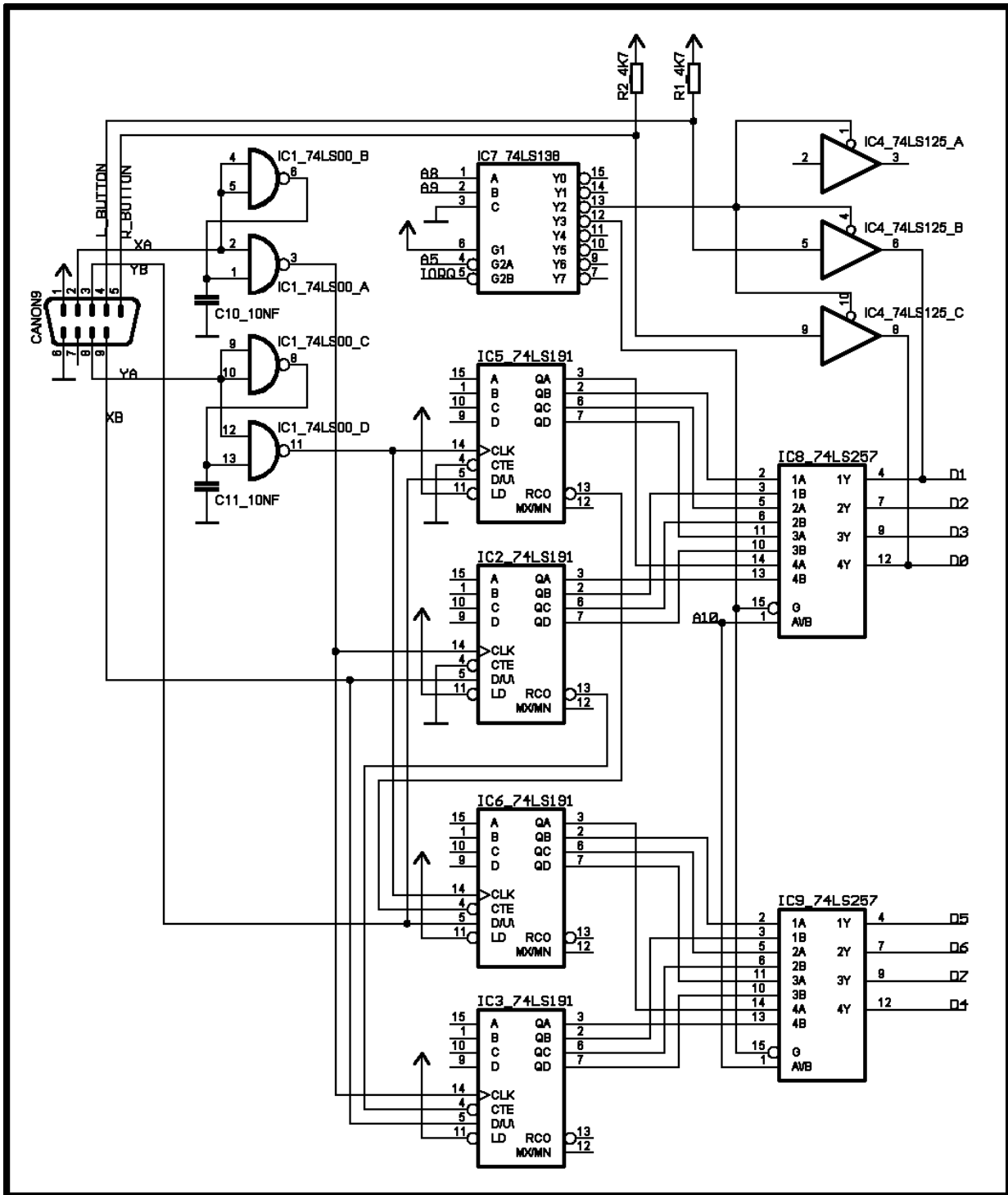
Fotka originál Kempston mouse interface s myší



Plošný spoj ze strany součástek a spojů



ZMAPOVANÉ SCHEMA ORIGINÁL KEMPSTON MOUSE INTERFACE



ORIGINAL KEMPSTON MOUSE INTERFACE

X-axis = port 64479 (xxxxx011 xx0xxxxx)
 Y-axis = port 65503 (xxxxx111 xx0xxxxx)
 BUTTONS= port 64223 (xxxxxx10 xx0xxxxx)

SOFTWARE PRO KEMPSTON MOUSE

Existuje několik originál her a programů, které podporovaly ovládání myši. Mezi ně patří například CARRIER COMMAND nebo BATTLE COMMAND. U poslední hry je zřejmě špatně napsaný ovladač a tak myš nefunguje. Také kreslicí program ARTIST 2 umí pracovat s myši. Pak je tu ještě spousta her a programů z Ruska, například HERO QUEST 1 a 2, SPACE CRUSADE, ARCANOID, GUNMAN, LAST COURIER 2, WOLF 2004, LEMMINGS 2, HEADBALL, MOORHUHN, SPARROWS, TECHNODROM, WALKER, FAST BREED, KAZMUS, SHASKI, KING, FARSPACE, a z programů třeba BURRIAL GRAPHIC EDITOR, různé TRACKERY, COMMANDERY a diskové magazíny. Nedávno jsem našel ještě ICON GRAPHIX 128. Nakonec jsou tu ještě moje předělávky her pro K-MOUSE. U některých her podporuji i 4. tlačítko myši, které umí obsloužit pouze K-MOUSE turbo interface. Všechny úpravy jsem řešil tak, aby bez jakéhokoli omezení mohli hru používat i majitelé třítlačítkové ruské Kempston myši, případně ji spouštět v emulátorech. Zatím jsem upravil jen několik her: R-TYPE, HEROQUEST 1 a 2, LEMMINGS 1, CHRONOS, SILKWORM, SIMCITY, YS CAPERS, ATOMIX, HEXAGONIA, DEFLEKTOR, SWIV a ZYNAPS, BATTLE COMMAND, OPERATION WOLF, MAIN BLOW, SLAP FIGHT a další. Většinou ve verzích 128 REMIX. U R-TYPE funguje levé tlačítko jako FIRE1, pravé jako FIRE2 a prostřední zapne/vypne autofire. Ve hře LEMMINGS 1(128remix) levé tlačítko je FIRE, pravým tlačítkem aktivujete sledování pozice lemminga, a prostřední tlačítko zruší funkci sledování. V SIMCITY má levé tlačítko funkci fire, prostřední = pohyb v menu nahoru, pravé = pohyb v menu dolů. Pokud máte vícetlačítkovou myš, stiskem 4. tlačítka vyvoláte mapu ostrova. Upravil jsem jak původní, tak českou(přeložil UNIVERSUM) verzi SIMCITY. U DEFLEKTORU levé a pravé tlačítko otáčí zrcadla. U her ATOMIX a HEXAGONIA je při držení tlačítka a současném pohybu myši emulován joystick.

EMULÁTORY ZX SPECTRA A KEMPSTON MOUSE

Dnes už každý pořádný emulátor ZX Spectra emuluje i Kempston mouse. Emulátor REALSPECTRUM je jeden z nejlepších, co se týče věrnosti emulace ZX, ale podporuje jen dvě tlačítka myši. Nejlepší zkušenosti mám s X128 a UNREALSPECTRUM, kde fungují již tři tlačítka. V UNREALSPECTRU bude nutné zapnout i funkci SWAP BUTTONS, která prohodí funkci levého a pravého tlačítka. Pak je tu emulátor ZX-EMUL, který podporuje i kolečko a je 100% kompatibilní s K-MOUSE TURBO. Tento emulátor najdete na <http://lion17home.narod.ru/> a na http://trd.speccy.cz/press/press_an.htm si pak stáhněte diskmagazíny DON NEWS, podporující od čísla 19 i kolečko myši. V DON NEWS 19 je schema zapojení kolečkové myši, zdrojáky ovladačů a tester. Nejkompletnější archiv her a programů s podporou Kempston mouse včetně dokumentace je na <http://www.velesoft.wz.cz>.

OVLAĐAČE PRO K-MOUSE

V podstatě nejrozšířenější verze ovladače:

```

;XECUTE MOUSE
;TAKE COORDINATES FROM CURPOS==COORD
MOUSE LD HL,(COORD)
      LD BC,#FBDF
      LD DE,(OLDKO)
      IN A,(C)
      LD (OLDKO),A
      SUB E
      JR Z,NM_X
      JP P,MX_PL
      ADD A,L
      JR C,ZER_X
      XOR A
ZER_X LD L,A
      JR NM_X
MX_PL ADD A,L
      JR C,BEX_Z
      CP #FD ;MAXIMUM X
      JR C,BEX_B
BEX_Z LD A,#FD ;MAXIMUM X
BEX_B LD L,A
NM_X  LD B,#FF
      IN A,(C)
      LD (OLDKO+1),A
      SUB D
      JR Z,NM_Y
      NEG
      JP P,MY_PL
      ADD A,H
      JR C,ZER_Y
      XOR A
ZER_Y LD H,A
      JR NM_Y
MY_PL ADD A,H
      JR C,BEY_Z
      CP #BE ;MAXIMUM Y
      JR C,BEY_B
BEY_Z LD A,#BE ;MAXIMUM Y
BEY_B LD H,A
NM_Y  LD A,H
      CP #FF
      JR C,BIGY
      LD H,#FF
BIGY  CP #02 ;MINIMUM Y
      JR NC,SMALY
      LD H,#02 ;MINIMUM Y
SMALY LD A,L
      CP #FF
      JR C,DIRY
      LD L,#FF
DIRY  CP #02 ;MINIMUM X
      JR NC,DIMENS
      LD L,#02 ;MINIMUM X
DIMENS LD (COORD),HL
      LD BC,#FADF
      LD HL,CONTRB
      IN A,(C)
      CPL
      AND 7
      RLCA
      RLCA
      RLCA
      RLCA
      OR (HL)

```

```

LD (HL),A
RET
;sem se ukladaji souradnice pro osu XaY
COORD DEFB 0,0
OLDKO  DEFB 0,0;WORKING

```

Jiná verze:

```

;MOUSE DRIVER WITH FIRE BUTTON
AUTOCONFIG
;(C) Andrey Rachkin'95
      JR      MDRV
DIRECTZ DEFB 0 ;FIRE
        DEFB 0 ;UP
        DEFB 0 ;DOWN
        DEFB 0 ;RIGHT
        DEFB 0 ;LEFT
        DEFB 0 ;CANCEL
MCOORD DEFW 0 ;LAST CURSOR COORDS
        ;IN PIXELZ
MPORTS DEFW 0 ;LAST READED MAUSY COORD
NONDEF AND 3 ;HERE COMEZ BUTTONZ
        ;CONTROL IF FIRE BUTON
        ;NOT DEFINED
      JR      Z,MDRV4 ;IF NONE BUTTON
        ;PUSHED
      CP      1
      JR      Z,NONDEF_
      XOR     A
      LD      (MDRV3+2),A
      LD      A,5
      LD      (MDRV2+2),A
NONDEF_ LD HL,0
        LD      (MDRV1),HL
        POP     IX
; ***** MAIN PROC OF MOUSEDRIVER *****
MDRV   PUSH   IX
        LD      HL,DIRECTZ
        PUSH   HL
        POP     IX
        XOR     A
        LD      (HL),A ;CLEARING
        INC     HL ;OF
        LD      (HL),A ;DIRECTZ
        INC     HL ;BUFER
        LD      (HL),A
        INC     HL
        LD      (HL),A
        INC     HL
        LD      (HL),A
        INC     HL
        LD      (HL),A
        INC     HL
        LD      BC,#FADF ;BUTTONZ CONTROL
        IN      A,(C) ;READ FROM PORT
        ;OF BUTTONS
      CPL
MDRV1  JR      NONDEF ;JR UNTIL FIRE
        ;BUTTON NOT DEFINED
      RRA
MDRV2  RL      (IX+0) ;FIRE
      RRA
MDRV3  RL      (IX+5) ;CANCEL
; ***** COORDS CONTROL *****
MDRV4  LD      HL,(MCOORD) ;FORM LAST

```

```

;CURSOR COORDS
LD DE,(MPORTS) ;FROM LAST
;READED MOUSE
;COORDS

LD BC,#FBDF
IN A,(C) ;READING FROM
;PORT X-COORD (0-#FF)
LD (MPORTS),A
SUB E
JR Z,MDRV9
JP P,MDRV6
LD (IX+4),1 ;MOVE LEFT
ADD A,L
JR C,MDRV5
XOR A ;MIN X-COORD
MDRV5 LD L,A
JR MDRV9
MDRV6 ADD A,L
LD (IX+3),1 ;MOVE RIGHT
JR C,MDRV7
CP #FE ;MAX X-COORD
JR C,MDRV8
MDRV7 LD A,#FE ;MAX X-COORD
MDRV8 LD L,A
MDRV9 LD B,#FF
IN A,(C) ;READING FROM
;PORT Y-COORD (0-#FF)
LD (MPORTS+1),A
SUB D
JR Z,MDRV14
NEG
JP P,MDRV11
LD (IX+1),1 ;MOVE UP
ADD A,H
JR C,MDRV10
XOR A ;MIN Y-COORD
MDRV10 LD H,A
JR MDRV14
MDRV11 ADD A,H
LD (IX+2),1 ;MOVE DOWN
JR C,MDRV12
CP #BF ;MAX Y-COORD
JR C,MDRV13
MDRV12 LD A,#BF ;MAX Y-COORD
MDRV13 LD H,A
MDRV14 LD (MCOORD),HL ;NEW CURSOR
;POSITION
;IN PIXELZ

POP IX
RET
; ***** DRIVER END *****

```

A další varianta s akcelerací pohybu:

```

; // Драйвер Кемпстон-мышы *** Kempston-mouse driver

OLDCO DEFW 0 ;(внутренняя переменная)
; координаты (x-младший, y-старший)
COORD DEFW 0
; три младших бита - кнопки (лев,прав,средн; 1-нажата)
CONTRB DEFB 0
MOUSE ;(c)1994-98 MaxRomanoff,Slava Mednonogov

; *****
LD BC,#FBDF
LD DE,(OLDCO)
LD HL,(COORD)
IN A,(C) ;вычисление X
LD (OLDCO),A

```

```

SUB E
JR Z,MSEx0
JP P,MSEx1
NEG
CALL MSElog
NEG
ADD A,L
JR C,MSExe
XOR A
JR MSExe
MSEx1 CALL MSElog
ADD A,L
JR C,MSEx11
CP 248 ;max X
JR C,MSExe
MSEx11 LD A,248 ;max X
MSExe LD L,A
MSEx0 LD B,#FF ;вычисление Y
IN A,(C)
LD (OLDCO+1),A
SUB D
JR Z,MSEy0
JP M,MSEy1
CALL MSElog
NEG
ADD A,H
JR C,MSEye
XOR A
JR MSEye
MSEy1 NEG
CALL MSElog
ADD A,H
JR C,MSEy11
CP 190 ;max Y
JR C,MSEye
MSEy11 LD A,190 ;max Y
MSEye LD H,A
MSEy0 LD (COORD),HL
LD B,#FA ;кнопки
IN A,(C)
LD (CONTRB),A
RET

MSElog CP 4 ;мышка с ускорением
RET C
ADD A,A
JR C,MSElgC
CP 8*2
RET C
ADD A,8
JR C,MSElgC
CP 120
RET C
MSElgC LD A,120
RET

```

а) Если акселератор мышы не нужен, удалите п/п ;MSElog и все обращения к ней.
б) Если минимальные координаты курсора больше, чем ;(0,0), необходимо ввести ;дополнительные ограничения

Následuje částečně okomentovaný zdrojový kód ovladače myši pro osu X,Y a tlačítka. Podobný ovladač jsem použil do všech mnou upravených her.

```
;OVLADAC K-MOUSE/K-MOUSE TURBO
```

```
;(MAXIMX+1) = maximum v ose X
;(MINIMX+1) = minimum v ose X
;(MINIMX2+1)= -II-
```

```
;(MAXIMY+1) = maximum v ose Y
;(MINIMY+1) = minimum v ose Y
;(MINIMY2+1)= -II-
```

```
;(COORD+0)= souradnice X sipky
;(COORD+1)= souradnice Y sipky
```

```
;(OLDCO+0)= stara osa X mysi
;(OLDCO+1)= stara osa Y mysi
```

```
;Pred prvnim spustenim ovladace
;je nutne spustit rutinku od
;adresy KOREKCE, ktera ovladaci
;podstrci aktualni souradnice
;mysi jako predchozi hodnotu.
;Tak zajisti, ze sipka bude po
;spusteni presne na vychozich
;souradnicich.
```

```
;ovladac lze volat dvojim zpuso
;bem. Pri volani od adresy
;KMNORMAL se testuje pouze osa
;X,Y a 3tlacitka mysi stejne
;jako u original K-MOUSE.
;Dalsi zpusob volani je od adr.
;KM-TURBO, kdy ovladac testuje
;osy X,Y a 4 tlacitka
;mysi.
```

```
;Ovladac vzdy uklada stav tlaci
;tek negovany, takze log.1 =
;stisknute tlacitko
```

```
;(BUTTONS+0) =
;D0=prave tlacitko
;D1=leve tlacitko
;D2=prostredni tlacitko
;D4=4.tlacitko(jen u KM-TURBO)
```

```
;v praxi staci jen spoustet pod
;prerusenim tento ovladac a pak
;z adresy COORD+0aCOORD+1 vycit
;at nove souradnice pro sipku.
;Ovladac jeste uklada primou
;adresu bajtu videoram, kam uka
;zuje souradnice sipky.
;Na adrese (KMVRAM)lezi 2B.adr.
;videoram pro vykresleni sipky.
;Na adrese (KMPIX)je pixelova
;pozice sipky(0-7)v danem bajtu
;A na adrese (KMATR)je 2B.adr.
;ukazujuci na odpovidajici mist
;o v atribut.casti videoram.
```

```
;vychozi souradnice sipky jsou
;pri prvnim spusteni ovladace
;ulozene na adresach COORD+0,+1
;=====
```

```
XMAX equ 255
XMIN equ 0
```

```
YMAX equ 191
YMIN equ 0
```

```
;start ovladace K-MOUSE
```

```
org 49152
```

```
START
```

```
;- - - - -
;tato rutina se vola pred prvni
;im spustenim ovladace. Slouzi
;pouze k vyrovnani odchylek
;mysi s vychozi polohou sipky.
```

```
KOREKCE
```

```
;ulozi aktualni osu X
ld a,%11111011
in a,(223)
ld (OLDCO),a
```

```
;ulozi aktualni osu Y
ld a,%11111111
in a,(223)
ld (OLDCO+1),a
ret
```

```
;- - - - -
;start ovladace pro puvodni
;K-MOUSE(osy X,Y a 3 tlacitka)
KMNORMAL
```

```
ld a,7
ld (BUTMASK+1),a
jr ORIGKM
```

```
;start ovladace pro novou K-MOU
;SE-TURBO(osy X,Y,4 tlacitka)
KMTURBO
```

```
ld a,15
ld (BUTMASK+1),a
```

```
ORIGKM
```

```
;osa X
MAXIMX ld a,XMAX
ld (SEM1+1),a
ld (BE_Z+1),a
xor a
ld (MODIFY),a
ld a,(COORD)
ld l,a
ld a,(OLDCO)
ld e,a
ld a,%11111011
in a,(223)
ld (OLDCO),a
call RUTINA
MINIMX cp XMIN
jr nc,SMALX
MINIMX2 ld l,XMIN
SMALX ld a,l
ld (COORD),a
ld a,h
ld (ODCH),a
```

```
;osa Y
```

```
MAXIMY ld a,YMAX
ld (SEM1+1),a
ld (BE_Z+1),a
ld a,237
ld (MODIFY),a
ld a,(COORD+1)
ld l,a
ld a,(OLDCO+1)
```

```

        ld     e,a
        ld     a,%11111111
        in     a,(223)
        ld     (OLDCO+1),a
        call  RUTINA
MINIMY   cp     YMIN
        jr     nc,SMALY
MINIMY2  ld     l,YMIN
SMALY    ld     a,l
        ld     (COORD+1),a
        ld     a,h
        ld     (ODCH+1),a
;tlacitka
        ld     a,%11111010
        in     a,(223)
        cpl
;podle rezimu prace maskuje jen
;3(KMNORMAL) nebo 4(KMTURBO)tlac
;itka mysi
BUTMASK  and    15
        ld     (BUTTONS),a

;nakonec se spocita adr.vram
;podle souradnic mysi
GO
        ld     a,(COORD+0)
        ld     l,a
        and    7
;ulozi posun v pixelech
        ld     (KMPIX),a

        ld     a,l
        srl   a
        srl   a
        srl   a
        ld     l,a
        ld     a,(COORD+1)
        rla
        rla
        and    %11100000
        or     1
;ulozi nizsi bajt adr.vram
        ld     (KMVRAM),a
;a jeste jednou. protoze nizsi
;bajt je shodny i pro adr.atrib
;utove casti vram
        ld     (KMATR),a

        ld     a,(COORD+1)
        and    7
        ld     l,a
        ld     a,(COORD+1)
        and    %11000000
        rrca
        rrca
        rrca
        or     64
        or     1
;ulozi vyssi bajt adr.vram
        ld     (KMVRAM+1),a

;nasleduje generovani adr.atri
;butu z adresy videoram
        rra
        rra
        rra
        and    %00000011
        or     %01011000
;uloz vyssi bajt adr.atributove
;casti vram
        ld     (KMATR+1),a

```

```

        ret
;-----
RUTINA
        sub    e
;odchylku zalohuj do H
        ld     h,a
;pri nulove odchylce preskoc
        jr     z,NM
;tady se meni NEG/NOP podle X/Y
MODIFY   neg
        jp     p,M_PL
        add    a,l
        jr     c,ZER
        xor    a
;posun souradnice sipky
ZER       ld     l,a
        jr     NM
M_PL     add    a,l
        jr     c,BE_Z
;MAX
;modifikuje 0 podle X,Y,K
SEM1     cp     0
        jr     c,BE_B
;modifikuje 0 podle X,Y,K
BE_Z     ld     a,0
BE_B     ld     l,a
NM       ld     a,l
        cp     255
        jr     c,BIG
        ld     l,255
BIG      ret
;-----
;predchozi souradnice X,Y
OLDCO    defb 0,0

;pozice sipky
COORD    defb 128,96

;odchylka v ose X,Y
ODCH     defb 0,0

;tlacitka
BUTTONS  defb 0

;adr.bajtu vram podle souradnic
KMVRAM   defw 0

;adr.atrib.casti vram
KMATR    defw 0

;posun sipky v pixelech vpravo
;muze byt v rozmezi 0-7
KMPIX    defb 0

```

Všechny dosud uvedené ovladače měly ošetřené maximum a minimum pro obě osy. Někdy ale tyto testy okrajů nejsou třeba, nebo jsou přímo nežádoucí. Pak si vystačíme s velice jednoduchým ovladačem. Dejme tomu máme šipku, kterou chceme pohybovat po celé ploše obrazovky, což je plocha o rozměru 256 (osa X) * 192 (osa Y). Při pohybu šipka může přelézat okraje obrazovky. Problém je jen v tom, že osa Y má u K-MOUSE stejný rozsah (0-255) jako osa X. Nejjednodušší řešení bude upravit rozsah osy Y z 0-255 na 0-191. A nějak takhle bych to řešil já:

```
KMDRV LD A,255
      IN A,(223)
      SRL A
      RL L
      SRL A
      RL L
      LD H,A
      RLCA
      ADD A,H
      LD H,A
      RR L
      JR NC,NEXT1
      INC H
NEXT1 RR L
      JR NC,OSA_Y
      INC H
OSA_Y LD A,251
      IN A,(223)
      LD L,A
BUTON LD A,250
      IN A,(223)
      CPL
      AND 7
```

Na výstupu je v reg.H nová osa Y v rozsahu 0-191 a v reg.L osa X (0-255). V registru A je na bitech D0-D2 stav tlačítek(bit v log.1=stisknuté) a současně je nastaven flag NZ (not zero), pokud je stisknuté jakékoli ze tří tlačítek. Jediná nevýhoda ovladače je v tom, že se změnou rozsahu osy Y se změní ve stejném poměru i rychlost pohybu kurzoru. Místo 4:4 bude poměr 4:3, ale věřte mi, že běžný uživatel nic nepozná. Ovladač může být ještě kratší, ale s omezením na velikost plochy 128 x 128. Jednoduše toho docílíme vydělením každé osy dvěma:

```
LD A,251
IN A,(223)
SRL A
LD L,A
LD A,255
IN A,(223)
SRL A
LD H,A
LD A,250
IN A,(223)
CPL
AND 7
```

Rychlost pohybu šipky se současně snížila v obou osách na polovinu.

OBSLUHA KOLEČKA MYŠI

Stav (poloha) kolečka je čitelný na bitech 4-7 portu tlačítek. Následující ovladač obsluhuje jen samotné kolečko a zdroják byl přepsán z diskmagazínu DON NEWS 19. Na adrese (W_COORD) je po návratu nová osa kolečka v rozsahu 0-255 bez ošetření maxima a minima. (Vedle obvodu PIC jsou dva prokovené otvory s nápisem "WHEEL", určené k osazení switche. Slouží k odpojení funkce kolečka myši)

```
WHEEL LD H,0
      CALL IN_WHEEL
      SUB H
      JR Z,S_DRIV+3
      JP M,MS
      CP 15-6
      JP NC,GET_UP1
      JP GET_DN
MS     NEG
      CP 15-6
      JP NC,GET_DN1
      JP GET_UP
GET_DN1 NEG
      AND #0F
GET_DN CALL W_SPEED
      LD H,A
      LD A,(W_COORD)
      SUB H
      JR S_DRIV
GET_UP1 NEG
      AND #0F
GET_UP CALL W_SPEED
      LD H,A
      LD A,(W_COORD)
      ADD A,H
S_DRIV LD (W_COORD),A
      RET
W_SPEED CP 1
      RET Z
      LD B,A
      LD E,A
      ADD A,E
      DJNZ $-1
      RET
IN_WHEEL LD BC,#FADF
      IN A,(C)
      AND #F0
      RRCA
      RRCA
      RRCA
      RRCA
      LD (WHEEL+1),A
      RET
W_COORD DEFB 0
```

Před prvním spuštěním doporučuji provést CALL IN_WHEEL. Tak se vyrovná odchylka polohy kolečka.

ÚPRAVA HER A PROGRAMŮ PRO K-MOUSE

Úprava programů pro K-MOUSE není nijak složitá. Dáme si za úkol například upravit nějakou 2D střílečku z bočního podhledu (CHRONOS nebo R-TYPE). V první řadě si zjistíme, kam si hra ukládá souřadnice (pozici) rakety a v jakém rozsahu se pohybují (maximum a minimum ve vertikálním a horizontálním směru). Najdeme si v paměti volné místo pro ovladač. Pak zjistíme adresu rutiny pro čtení klávesnice nebo joysticku a zmapujeme si její funkci. Tyto rutiny většinou vrací stav o směru pobytu a fire jako jednobajtovou hodnotu v nějakém registru. Pro čtení kempston joysticku najdeme často něco podobného:

```
#C000 LD BC,#1F
#C003 IN A,(C)
#C005 LD (#5B00),A
atd.....
```

Tady je ideální místo čtení joysticku volat ovladač myši s testem tlačítek. Ten po návratu vrátí v registru A hodnotu 16 při stisknutém tlačítku, nebo 0 při nestisknutém. Od adr. #C000 přepíšeme kód na:

```
#C000 CALL MFIRE
#C003 NOP
#C004 NOP
#C005 LD (#5B00),A
```

Tak zajistíme přesměrování na test tlačítka, jehož kód můžeme napsat nějak takto:

```
MFIRE LD A,#FA
;A=stav tlačítek myši
IN A,(#DF)
;testujeme jen levé tlačítko
BIT 1,A
LD A,#00
;pokud není stisk, vrátí A=0
JR NZ,END
;stisk emuluje stav fire joysticku
FIRE LD A,#10
END RET
```

Ještě si vyhledáme rutinu, která mění souřadnice podle původně načtené hodnoty z portu joysticku. To bude vypadat asi jakto:

```
#8200 LD A,(#5B00) ;vzvedne stav joysticku
#8203 LD B,A
#8204 RR B
#8205 CALL NZ,#8255 ;odskok při směru
vpravo
#8208 RR B
#8209 CALL NZ,#8289 ;odskok při směru vlevo
atd....
```

Na adrese #8255 musí být něco jako:

```
LD A,(#6253) ;protože se tato rutina
```

```
; volá pro směr vpravo, je jasné, že je na této
; adrese osa X
CP #CF ;test maxima v ose X
;návrat, když je pozice na maximu
RET Z
INC A
;posune souřadnici a uloží zpět
LD (#6253),A
RET
```

V předchozím případě, při přesměrování na test fire, nám rutina vrací kromě bitu 4 (fire) všechny bity v log.0. Takže program nikdy nebude detekovat směry joysticku. Toho hned využijeme. Na adrese #8200 přepíšeme první instrukci na CALL MOUSEDRV (volání ovladače myši pro osu X a Y). Adresa MOUSEDRV ukazuje na volné místo v ram, kam umístíme ovladač. Na začátku ovladače si zazálohujeme všechny registry, které budeme používat, pak vložíme vlastní ovladač, na jehož konci těsně před návratem opět zazálohované registry obnovíme.

Ovladač myši:

```
MOUSEDRV PUSH AF
PUSH HL
PUSH DE
```

Následuje kód ovladače

```
RETMDRV POP DE
POP HL
POP AF
LD A,(#6253) ; Před návratem
;vložíme do A opět stav joysticku (vlastně jen fire)
RET
```

V ovladači nastavíme správně rozsahy pro osy X, Y a adresy, kam si hra ukládá souřadnice. Teď už by nám mělo fungovat ovládání souřadnic a fire pomocí myši. U některých her stačí jen místo instrukce pro čtení portu joysticku vložit jediný CALL našeho ovladače. Ten rovnou změní souřadnice, a v reg. A vrátí stav fire. Ne každá hra se s tím ale vyrovná, tak je lepší volat ovladač pro osy až přímo na začátku rutiny, která původně souřadnice měnila. V úvodním menu si zjistíme, kam si hra ukládá zvolený typ ovládání a na začátek ovladače přidáme ještě test, jestli je nastavený kempston. Pokud bude ovládání jiné, ovladač se rovnou vrátí, aniž by měnil souřadnice (před návratem musíme provést ještě instrukci LD A,(#5B00), kterou jsme přepsali). Nakonec ještě přepíšeme v menu název „KEMPSTON“ na „K-MOUSE_“.

ZDROJOVÝ KÓD K-MOUSE TURBO

Následující zdrojový kód pro XILINX XC9572XL-PC44 je psaný v jazyce ABEL

```
MODULE KMLAST9572
// 4.8.2005 VELESOFT

//plna verze pro externi pruchozni interface s CPLD - XILINX XC9572XL PC44
//u vsehch portu je pouzita uplna adresace !!!

//Upraveny rovnice hodin(CLK) pro ukladani stavu clonek mysi(BXA,BXB,BYA,BYB)
//Orezane citace os X a Y na 8 bitu => pri prepnuti rychlosti mysi se nemeni
//souradnice(drive posun dat na 10 bit citacich)
//Diky novemu CLK a kratsim citacum usetreny 4 registry a hromada termu !
//Po zapnuti je stav citace kolecka 1111(misto puvodni 0000). Software
//spolehajici se na hodnotu D7-D4 v log.1(na portu tlacitek) tak bude
//fungovat baz nutnosti kolecko vypinat switchem. Ovsem za predpokladu,
//ze se koleckem nebude hybat.

//opetovny trojstisk v extra modu resetuje nastaveni mysi

//porty 31(AMIGA MOUSE/KEMPSTON JOY) se povoluji switchem ENABLE31
//switch AMOUSE nastavi emulaci amiga mouse na portu 31. IN 31=0 pri SWITCH OFF.

//na D4-D7 lezi 4 bitovy citac polohy kolecka. Zapojeni bylo prevzato z
//DON-NEWS19(donnws19).

//pin c.34 vypina 4.bitovy citac kolecka mysi na portu tlacitek.

//Adresace portu K-MOUSE, A-MOUSE
//tlacitka = 64223 xxxxx0x0 11011111(na D4-D7 je 4-bitovy citac pro kolecko)
//osa X = 64479 xxxxx0x1 11011111
//osa Y = 65503 xxxxx1x1 11011111
//A-MOUSE/KEMPSTON JOY = 31 00011111

//K-MOUSE: podpora ctyr tlacitek + kolecka mysi
//port tlacitek c. 64223:
// D0-prave tlacitko
// D1-leve tlacitko
// D2-prostredni tlacitko
// D3-4.tlacitko
// D4-bit 0 citace kolecka
// D5-bit 1 citace kolecka
// D6-bit 2 citace kolecka
// D7-bit 3 citace kolecka
//(stisknute tlacitko = log.0, nestisknute = log.1)

//A-MOUSE a KEMPSTON JOYSTICK: port 31
//D0-XB / joy RIGHT
//D1-YB / joy LEFT
//D2-XA / joy DOWN
//D3-YA / joy UP
//D4-leve tlacitko / joy FIRE 1
//D5-prave tlacitko / joy FIRE 2
//D6-prostredni tlacitko / joy FIRE 3
//D7-ctrvrte tlacitko
//(aktivni smer/fire v log.1)

////////////////////////////////////

//VSTUPNI PINY

DISWHEEL PIN 34;//log.0 vypne citac kolecka na portu tlacitek(64223)
DISEXTRA PIN 19;//BLOKUJE EXTRA MOD(prepinac v log.0)
ENABLE31 PIN 20;//pravý switch zprístupni port 31(v tomto režimu je možné
//stiskem FIRE na joysticku odpojit emulaci A-MOUSE a na
//portu 31 bude citelny KEMPSTON JOY
//po stisku tlacitka mysi se opet odpoji JOYSTICK a bude
//citelna A-MOUSE
AMOUSE PIN 18;//levý switch(na desce znacen jako AMOUSE)zprístupni na
//portu 31 AMIGA MOUSE.(pak lze nulovat port 31 dodatecne
//z extra modu). Nezapojeny switch = port 31 tvrde nulovan.
XENABLE31 NODE ISTYPE 'REG'; //trojstisk,LEFT BUTON - vyp/zap portu 31
//((nuluje port/emulace A-MOUSE)
STISK NODE istype 'reg'; //reg.detekce trojstisku - zapne extra mod a
```

```

//tlacitka nejsou citelna
POVOL      NODE istype 'reg'; //reg.detekce uvolneni trojstisku
CLEAR      NODE istype 'reg'; //reg.nulujici STISK a POVOL pri opusteni modu
EXBUT      NODE istype 'reg'; //reg.stav layoutu tlacitek(EXBUT/NORMAL)

//registry s predchozim stavem clonek mysi v ose X
BXA      NODE istype 'reg';
BXB      NODE istype 'reg';
BX = [BXA,BXB];

//registry s predchozim stavem clonek mysi v ose Y
BYA      NODE istype 'reg';
BYB      NODE istype 'reg';
BY = [BYA,BYB];

//signaly smeru pri pohybu mysi
LEFT     NODE istype 'COM';
RIGHT    NODE istype 'COM';
UP       NODE istype 'COM';
DOWN    NODE istype 'COM';

//adresy z CPU
A0      PIN 8;
A1      PIN 6;
A2      PIN 4;
A3      PIN 2;
A4      PIN 39;
A5      PIN 40;
A6      PIN 42;
A7      PIN 43;
A8      PIN 38;
A10     PIN 37;

//ridici signaly z CPU
IORQ    PIN 1;
RD      PIN 44;

//signaly z A-MOUSE(PICu)
XA      PIN 26; //joy DOWN
XB      PIN 24; //joy RIGHT
YA      PIN 22; //joy UP
YB      PIN 25; //joy LEFT
RIGHTB  PIN 35; //joy FIRE 2
LEFTB   PIN 33; //joy FIRE 1
MIDDLEB PIN 36; //joy FIRE 3
FIRE4   PIN 27; //4.tlacitko mysi

//nove vstupy pro clonky kolecka
KA      PIN 29;
KB      PIN 28;

KUP     NODE istype 'COM';
KDOWN   NODE istype 'COM';
KYA     NODE istype 'REG';
KYB     NODE istype 'REG';
KY = [KYA,KYB];

K3..K0  NODE ISTYPE 'REG'; //citac kolecka je jen 4-bitovy
KCOUNTER = [K3..K0];

//VYSTUPNI PINY
D7      PIN 14 istype 'com';
D6      PIN 9  istype 'com';
D5      PIN 7  istype 'com';
D4      PIN 3  istype 'com';
D3      PIN 5  istype 'com';
D2      PIN 11 istype 'com';
D1      PIN 12 istype 'com';
D0      PIN 13 istype 'com';
DATA    = [D7..D0];

//INTERNI VYSTUPY
XQ9..XQ0 NODE istype 'reg';
XCOUNTER = [XQ9..XQ0]; //citac pro osu X
YQ9..YQ0 NODE istype 'reg';
YCOUNTER = [YQ9..YQ0]; //citac pro osu Y

MSLOW   NODE ISTYPE 'REG'; //reg.rychlosti pohybu FAST/SLOW
SETUPRESET NODE ISTYPE 'REG'; //registr pro reset vlastnosti interface

```

////////////////////////////////////

EQUATIONS

//smery pohybu mysí

```
RIGHT = !BXA & !BXB & XA & !XB
# BXA & !BXB & XA & XB & !MSLOW //MSLOW v log.0 = turbo pohyb 4 x, log.1 = slow
# BXA & BXB & !XA & XB & !MSLOW
# !BXA & BXB & !XA & !XB & !MSLOW;
```

```
LEFT = !BXA & !BXB & !XA & XB
# !BXA & BXB & XA & XB & !MSLOW
# BXA & BXB & XA & !XB & !MSLOW
# BXA & !BXB & !XA & !XB & !MSLOW;
```

```
DOWN = !BYA & !BYB & YA & !YB
# BYA & !BYB & YA & YB & !MSLOW
# BYA & BYB & !YA & YB & !MSLOW
# !BYA & BYB & !YA & !YB & !MSLOW;
```

```
UP = !BYA & !BYB & !YA & YB
# !BYA & BYB & YA & YB & !MSLOW
# BYA & BYB & YA & !YB & !MSLOW
# BYA & !BYB & !YA & !YB & !MSLOW;
```

//pri pohybu mysí uchova aktualni stav clonek

```
BXA := XA;
BXB := XB;
BX.clk = !BXA & !BXB & XA & !XB //V BXA a BXB je predchozi stav clonek
# BXA & !BXB & XA & XB //Pokud predchozi stavy clonek nesouhlasí s aktualnim,
# BXA & BXB & !XA & XB //ulozi aktualni stav do BXA a BXB.
# !BXA & BXB & !XA & !XB
# !BXA & !BXB & !XA & XB
# !BXA & BXB & XA & XB
# BXA & BXB & XA & !XB
# BXA & !BXB & !XA & !XB;
```

BYA := YA;

BYB := YB;

```
BY.clk = !BYA & !BYB & YA & !YB
# BYA & !BYB & YA & YB
# BYA & BYB & !YA & YB
# !BYA & BYB & !YA & !YB
# !BYA & !BYB & !YA & YB
# !BYA & BYB & YA & YB
# BYA & BYB & YA & !YB
# BYA & !BYB & !YA & !YB;
```

//soucasny stisk trech tlacitek mysí aktivuje extra mod

```
STISK.clk = !LEFTB & !MIDDLEB & !RIGHTB & !STISK & !DISEXTRA;
STISK := 1;
STISK.ar = CLEAR; //nuluje pri opusteni extra modu
```

//po uvolneni trojstisku se ceka na dalsi stisk tlacitka

```
POVOL.clk = LEFTB & RIGHTB & MIDDLEB & STISK;
POVOL := 1;
POVOL.ar = CLEAR; //nuluje pri opusteni extra modu
```

//opusteni extra modu stiskem a povolenim jakéhokoli tlacitka

```
!CLEAR.CLK = STISK & POVOL & !LEFTB
# STISK & POVOL & !RIGHTB
# STISK & POVOL & !MIDDLEB;
CLEAR := 1;
CLEAR.ar = !STISK & !POVOL;
```

SETUPRESET.CLK = STISK & POVOL & !LEFTB & !RIGHTB & !MIDDLEB;

//v extra modu stisk LB+RB+MB resetuje nastaveni mysí

```
SETUPRESET := 1;
SETUPRESET.AR = !STISK;
```

//trojstisk(extra mod)+prave tlac.=prohozeni leveho a praveho tlac.

```
EXBUT.clk = STISK & POVOL & !LEFTB & !MIDDLEB & !RIGHTB;
EXBUT := !EXBUT;
//opetovny trojstisk v extra modu resetuje nastaveni mysí.
EXBUT.ar = SETUPRESET;
```

//trojstisk+leve tlac. = funkce zap/vyp nulovani portu 31 je pristupna

//jen pokud je zapojen switch AMOUSE.

```
XENABLE31.clk = STISK & POVOL & !LEFTB & !MIDDLEB & !RIGHTB & !AMOUSE;
XENABLE31 := !XENABLE31;
//opetovny trojstisk v extra modu resetuje nastaveni mysí.
XENABLE31.ar = SETUPRESET;
```

```

//trojstisk+prostredni tlac. = rychlost pohybu SLOW / TURBO 4x
!MSLOW.CLK      = STISK & POVOL & LEFTB & !MIDDLEB & RIGHTB;
MSLOW          := !MSLOW;
//opetovny trojstisk v extra modu resetuje nastaveni mysi.
MSLOW.ar       = SETUPRESET;

//clock pro citace os X a Y
XCOUNTER.clk   = LEFT & !RIGHT
               # RIGHT & !LEFT;
YCOUNTER.clk   = UP & !DOWN
               # DOWN & !UP;

//citac pro osu X
WHEN (!LEFT & RIGHT) //pohyb vpravo
THEN XCOUNTER := XCOUNTER + 1
WHEN (!RIGHT & LEFT) //pohyb vlevo
THEN XCOUNTER := XCOUNTER - 1

//citac pro osu Y
WHEN (!DOWN & UP) //pohyb nahoru
THEN YCOUNTER := YCOUNTER + 1
WHEN (!UP & DOWN) //pohyb dolu
THEN YCOUNTER := YCOUNTER - 1

//Pri extra modu nejsou citelna tlacitka K-MOUSE a port 31 je nulovan !!!
//Output enable pro datove signaly
DATA.oe = !A10 & A7 & A6 & !A5 & A4 & A3 & A2 & A1 & A0 & !RD & !IORQ
//port 64479 - osa X + 64223 - tlacitka(+vypinatelne kolecko)
           # A10 & A8 & A7 & A6 & !A5 & A4 & A3 & A2 & A1 & A0 & !RD & !IORQ
//port 65503 - osa Y
           # !ENABLE31 & !A7 & !A6 & !A5 & A4 & A3 & A2 & A1 & A0 & !RD & !IORQ;
//PORT 31 (A-MOUSE/KEMPSTON JOY)

//multiplex dat. Prepina 3x port K-MOUSE,1x port A-MOUSE
D0      = !A10 & !A8 & A7 & A1 & LEFTB & EXBUT //po prohozeni L/R tlacitek je zde leve tlac.
         # !A10 & !A8 & A7 & A1 & RIGHTB & !EXBUT //defaultne je na D0 portu 64223 prave tlac.
         # !A10 & !A8 & A7 & A1 & STISK//blokuje cteni tlacitek v EXTRA MODU
         # !A10 & A8 & A7 & A1 & XQ0 //data osy X
         # A10 & A8 & A7 & A1 & YQ0 //data osy Y
         # !A7 & !A6 & A1 & !XB & !XENABLE31 & !STISK & !AMOUSE;//AMIGA-MOUSE pristupna jen pokud
                                                //je switch AMOUSE zapojeny.

D1      = !A10 & !A8 & A7 & A1 & RIGHTB & EXBUT
         # !A10 & !A8 & A7 & A1 & LEFTB & !EXBUT
         # !A10 & !A8 & A7 & A1 & STISK
         # !A10 & A8 & A7 & A1 & XQ1
         # A10 & A8 & A7 & A1 & YQ1
         # !A7 & !A6 & A1 & !YB & !XENABLE31 & !STISK & !AMOUSE;

D2      = !A10 & !A8 & A7 & A1 & MIDDLEB
         # !A10 & !A8 & A7 & A1 & STISK
         # !A10 & A8 & A7 & A1 & XQ2
         # A10 & A8 & A7 & A1 & YQ2
         # !A7 & !A6 & A1 & !XA & !XENABLE31 & !STISK & !AMOUSE;

D3      = !A10 & !A8 & A7 & A1 & FIRE4
         # !A10 & !A8 & A7 & A1 & STISK
         # !A10 & A8 & A7 & A1 & XQ3
         # A10 & A8 & A7 & A1 & YQ3
         # !A7 & !A6 & A1 & !YA & !XENABLE31 & !STISK & !AMOUSE;

D4      = !A10 & !A8 & A7 & A1 & (!K0 # !DISWHEEL)//na D4-D7 je 4-bitovy citac polohy kolecka
         # !A10 & A8 & A7 & A1 & XQ4
         # A10 & A8 & A7 & A1 & YQ4
         # !A7 & !A6 & A1 & !LEFTB & !XENABLE31 & !STISK & !AMOUSE;

D5      = !A10 & !A8 & A7 & A1 & (!K1 # !DISWHEEL)
         # !A10 & A8 & A7 & A1 & XQ5
         # A10 & A8 & A7 & A1 & YQ5
         # !A7 & !A6 & A1 & !RIGHTB & !XENABLE31 & !STISK & !AMOUSE;

D6      = !A10 & !A8 & A7 & A1 & (!K2 # !DISWHEEL)
         # !A10 & A8 & A7 & A1 & XQ6
         # A10 & A8 & A7 & A1 & YQ6
         # !A7 & !A6 & A1 & !MIDDLEB & !XENABLE31 & !STISK & !AMOUSE;

D7      = !A10 & !A8 & A7 & A1 & (!K3 # !DISWHEEL)
         # !A10 & A8 & A7 & A1 & XQ7
         # A10 & A8 & A7 & A1 & YQ7
         # !A7 & !A6 & A1 & !FIRE4 & !XENABLE31 & !STISK & !AMOUSE;//ctvrte tlacitko pro A-MOUSE

// OBSLUHA KOLECKA

```

```

// Na datech D4-D7 je citelny negovany stav kolecka, takze vycchozi stav je 1111.
//KYA a KYB je pro uchovani predchoziho stavu clonek kolecka
KYA := KA;
KYB := KB;
//a uklada se pri kazdem pohybu kolecka
KY.clk = KUP & !KDOWN
      # KDOWN & !KUP;

//generovani smeru pohybu kolecka mysi
KUP    =(!KYA & !KYB & KA & !KB
      # KYA & !KYB & KA & KB
      # KYA & KYB & !KA & KB
      # !KYA & KYB & !KA & !KB);

KDOWN  =(!KYA & !KYB & !KA & KB
      # !KYA & KYB & KA & KB
      # KYA & KYB & KA & !KB
      # KYA & !KYB & !KA & !KB);

//hodiny citace pohybu kolecka
KCOUNTER.CLK = KUP & !KDOWN
      # KDOWN & !KUP;

WHEN (KUP & !KDOWN)
THEN KCOUNTER := KCOUNTER+1 //(puvodne -)
ELSE
WHEN (KDOWN & !KUP)
THEN KCOUNTER := KCOUNTER-1 //(puvodne +)

END

```