EASY 68K

[EASy68K Examples](http://www.easy68k.com/easy68kexamples.htm)

[68000 (blackboardcdn.com)](https://learn-eu-central-1-prod-fleet01-xythos.content.blackboardcdn.com/60d4531e78936/5222678?X-Blackboard-S3-Bucket=learn-eu-central-1-prod-fleet01-xythos&X-Blackboard-Expiration=1710439200000&X-Blackboard-Signature=C%2B7L1tvoqNjTSGBCFMO5x0bKQz3b8%2Fx84O43pKO%2BoWw%3D&X-Blackboard-Client-Id=680538&X-Blackboard-S3-Region=eu-central-1&response-cache-control=private%2C%20max-age%3D21600&response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%2768000.pdf&response-content-type=application%2Fpdf&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEG0aDGV1LWNlbnRyYWwtMSJHMEUCIEieiroK4vXEDqgWvFlXM8HU6PnqqgeZVF4u7Yaba71YAiEAi9nhmVsTu%2BiWvTB673SnzouBFdGoK76PHW205zcYyH4qvgUIdhADGgw2MzU1Njc5MjQxODMiDBd2We0QUkjksOnv1SqbBfVghfrqmN72cxIJNKtZMQ4TcpI%2F6q3absFxKeGxhjRPnPccJQX%2BDF8g1h1lyGH8CXTxVkR97sT2xtbJyjAqE1vOgAfJ1RuzHiPcPXE5MBrlpymW7D0R1kXtL7IX77Wg1DsnHJiFKCQLpcXzY5MGiaBs1i6r8UIa%2FVN9yeUvoqsBasS6FRlh0k2pYTWe7FBVQq%2BqzpB7K%2BZf7u3PpM5tUWh3Rtm5pFPixA2UMHJ0eCWEdbKJxcmK80zoWch6PFwymdhZ4hgpTUG%2BOHdIDp6ednwdjhxs5yGYFY2cC2xGBlkAYpTloTQuxkppIGDdLkxvleWRPlW47OE8AN9DR%2BB4e7NUQBXTKyJvT1h7E6T%2FDgTmARoupnivqwmq9F6fMQZR6miajTXiXAMekRYsyl4%2BobJ0mPo6Gr9h8ky5%2Fk%2FCAeg%2F%2Bk4JPiIDlspqdA%2Fs7jZZpFeTYnuJgXLRI%2FvqRnU1%2FLWEyk5X5hLlRvaTDJ5rLq4pP2TGt5J1JIKjTKOA%2F2dTkuJeJrPrnuxI2dW0M4zm7BE%2FTXpXKwZVwuYMTbM4crdKgODoqCIMw%2BpqLOkLm27d3epEfyeGLV3KyFndSJeTuSOzIDne92G1hrtQOlbi4Y6bwF24HUBlBIM37v5O%2BtkeJibb1uhxCTMvxNagoK8SwVXQULbl6yzK%2BvP4W3KKNWwRdqHJ3OOmtyTAYioE56x%2FrTqCjdJl6Gtl4DOle61xJWg0q3d3tvE8Ii80nBmKvQR0ahuBD1M%2BG7a%2FF2pZuunutwxSV75ygxycptYsWH0fk68WrrafN8I5SxdPyGc%2FStuU8bOxYZAXTJewvtDG03HideAxfkYOdkE491OJV8EDlK7yw64gOMlfzInjW5S2mlXzrm7%2FzMJuzk5eiHww9O7LrwY6sQFKwyYCOFfOrBpqurPCanYUq8nQ5j%2Fu47EQyxIgkSftGAqEVW4rdtIbU7hfpcwVtIxxqK35McvEZzvTX7ij30%2By8BFY9CyRGHKwDQ1IkInJL5vH7wvFSO%2BFLQzZx%2FqUoho5tN%2Fnwdmv6uzlICNV1yJU0pka9uU%2B9Bnh6BkQWWYMTpYpL5yOP%2Ft21x29gmyrdE2QH5sS2uI4bvJNTSHnfoEiigiJfosxvPhjHJaUVI3FToU%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20240314T120000Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=ASIAZH6WM4PLQM2CJPMZ%2F20240314%2Feu-cen)

[Text I/O (easy68k.com)](http://www.easy68k.com/QuickStart/TrapTasks.htm)

[Paper Title (use style: paper title) (ripublication.com)](https://www.ripublication.com/ijeet19/ijeetv9n2_03.pdf)

draw\_line

bsr grey

bsr points

move.l #84,d0 ////draws line

trap #15

bsr blacky

rts

draw\_square

bsr grey

bsr points

move.l #87,d0 ////draws square

trap #15

bsr blacky

rts

draw\_circle

bsr grey

bsr points

move.l #88,d0 ////draws circle

trap #15

bsr blacky

rts

LAB\_initl

MOVEQ #1,d1 \* set background colour

MOVEQ #81,d0 \* set fill colour

TRAP #15

LAB\_init\_disp

MOVEQ #5,d7 \* six digits to do

MOVEQ #0,d1 \* set pen black

MOVEQ #80,d0 \* set pen colour

TRAP #15

\* Make sure "Enable Exceptions" is checked in the simulator Options menu.

\*

\* Trap #15 Task 60 Enable/Disable mouse IRQ

\* An IRQ is created when a mouse button is pressed, released or moved in the output window.

\* D1.W High Byte = IRQ level (1-7), 0 to turn off

\* D1.W Low Byte = Mouse event that triggers IRQ:

\* Bit2 = Move, Bit1 = Button Up, Bit0 = Button Down

\* (Example D1.W = $0103, Enable mouse IRQ level 1 for Move, Button Up and Button Down)

\* (Example D1.W - $0002, Disable mouse IRQ for Button Up)

\*

\* Trap #15 Task 61 reads mouse

\* D1.B = 00 to read current state of mouse

\* = 01 to read mouse up state

\* = 02 to read mouse down state

\* The mouse data is contained in the following registers

\* D0 as bits = Ctrl, Alt, Shift, Double, Middle, Right, Left

\* Left is Bit0, Right is Bit 1 etc.

\* 1 = true, 0 = false

\* Shift, Alt, Ctrl represent the state of the corresponding keys.

\* D1.L = 16 bits Y, 16 bits X in pixel coordinates. (0,0 is top left)

\*

\* Trap #15 Task 33 - Get/Set Output Window Size

\* D1.L High 16 bits = Width in pixels, min = 640

\* Low 16 bits = Height in pixels, min = 480

\* D1.L = 0, get current window size as

\* High 16 bits = Width

\* Low 16 bits = Height

; Initialize IRQ vectors

move.l #mouseDownIRQ,$64 ; IRQ1

move.l #mouseUpIRQ,$68 ; IRQ2

move.l #mouseMoveIRQ,$6C ; IRQ3

andi.w #$00,SR ; put CPU in User mode

move.b #60,d0 ; set mouse IRQ

move.w #$0101,d1 ; IRQ1 for mouse down

trap #15

move.b #60,d0 ; set mouse IRQ

move.w #$0202,d1 ; IRQ2 for mouse up

trap #15

move.b #60,d0 ; set mouse IRQ

move.w #$0304,d1 ; IRQ3 for mouse move

trap #15

move.b #33,d0 ; set window size

move.l #(800<<16+600),d1 ; 800 x 600

trap #15

\* IRQ handlers

; mouse down handler

mouseDownIRQ

movem.l d0-d1,-(a7)

bsr heading

move.b #61,d0 ; read mouse

move.b #2,d1 ; mouse down state

trap #15

bsr showMouseData

movem.l (a7)+,d0-d1

rte

; mouse up handler

mouseUpIRQ

movem.l d0-d1,-(a7)

bsr heading

move.b #61,d0 ; read mouse

move.b #1,d1 ; mouse up state

trap #15

bsr showMouseData

movem.l (a7)+,d0-d1

rte

; mouse move handler

mouseMoveIRQ

movem.l d0-d1,-(a7)

bsr heading

move.b #61,d0 ; read mouse

move.b #0,d1 ; current mouse state

trap #15

bsr showMouseData

movem.l (a7)+,d0-d1

rte

\* Pre: D0 & D1 contain mouse data

showMouseData

movem.l d0-d4/a1,-(a7)

move.l d0,d4 ; save mouse buttons

bsr cursor ; draw cursor

move.l d1,d3 ; save mouse Y

and.l #$0000FFFF,d1 ; remove Y from D1

move.b #6,d2 ; field width 6

move.b #20,d0 ; display signed number in D1.L in field D2.B columns wide

trap #15 ; dislay X

move.l d3,d1

swap d1

and.l #$0000FFFF,d1 ; mouse Y

trap #15 ; display Y

move.b #14,d0 ; task 14, display string without CR,LF

for.b d3 = #1 to #7 do.s

lsr.b #1,d4 ; check mouse data bit in d4

if <cs> then.s ; if set

lea strYes,a1 ; get Yes message

else

lea strNo,a1 ; get No message

endi

trap #15 ; display

endf

movem.l (a7)+,d0-d4/a1

rts