

World RAM

This is a list of the RAM locations used by the world map program, including chocobo, airship, and serpent trench. The code that uses this data is in bank EE.

\$0000-\$00FF: World Direct Page

```
+$00 always #$0000
+$02 always #$FFFF
+$04 unmodified buttons pressed this frame
+$06 buttons pressed this frame modified based on button config
+$08 buttons pressed this frame but not last frame
+$0A -
+$0C buttons pressed last frame
    axlr----
    a: A button down
    x: X button down
    l: L button down
    r: R button down
    byetudlr
    b: B button down
    y: Y button down
    e: Select button down
    t: Start button down
    u: Up direction down
    d: Down direction down
    l: Left direction down
    r: Right direction down

$19

+$1E
$20

$22 target screen brightness
$23 current screen brightness (0-F)
$24 vblank flag
+$26 forward movement speed
++$29 rotation speed
+$2F altitude
$31

+$34 current X position (in pixels)
+$38 current Y position (in pixels)
++$3A rotation angle (high 16-bits active)
+$3D m7a
+$3F m7b
+$41 m7c
```

```
+$43 m7d
+$44
+$46
$58-$64 scratchpad

$60 fdd?????
f: vehicle is moving forward
d: double speed turns ???
$61 ???dulr
d: vehicle is moving down
u: vehicle is moving up
l: vehicle is turning left
r: vehicle is turning right
+$6A scratchpad

+$73 rotation frame counter
+$75 vehicle direction ???
+$77 bg1 horizontal scroll
+$79 bg1 vertical scroll
+$7B map horizontal scroll position, or maybe an offset based on the map
sector ???
+$7D map vertical scroll position ???
+$83
+$85
+$87
+$89
$8B
$8C zoom level ??
$9F HDMA #4 (M7A) line count 1 (124 lines, no repeat)
+$A0 HDMA #4 (M7A) data pointer 1
$A2 HDMA #4 (M7A) line count 2 (100 lines, no repeat)
+$A3 HDMA #4 (M7A) data pointer 2
$A5 HDMA #4 (M7A) line count 3 (terminate)
$A6 HDMA #5 (M7B) line count 1 (124 lines, no repeat)

$AD HDMA #6 (M7C) line count 1 (124 lines, no repeat)

$B4 HDMA #7 (M7D) line count 1 (124 lines, no repeat)

$C2 ?bfpssac current tile properties, byte 1
b: enable battles
f: forest
p: passable on foot
s: airship shadow size
a: airship can land
c: passable with chocobo
$C3 kpv??bbb current tile properties, byte 2
k: kefka's tower
p: phoenix cave
v: veldt
```

```

    b: battle background
+$C4 current tile index
+$C6 current X position (in pixels)
+$C8 current Y position (in pixels)
    $CA character 1 graphic (00 = none, 01 = falcon, 02 = chocobo ???, 03 =
character, 04 = ???, 05 = ship, 0C = esper terra, 12 = bird, 16 = smoking
airship)
    $CB character 2 graphic
    $CC character 3 graphic
    $CD character 4 graphic
++$D2 decompression source address
++$D5 decompression destination address
+$DF X position (in pixels * 256)
+$E1 Y position (in pixels * 256)
+$E3 X movement speed (signed)
+$E5 Y movement speed (signed)
    $E7 ??st?wvc
        s: character/vehicle is shown
        t: character is transparent (bottom sprite, for forests)
        w: character/vehicle is moving/waiting
        v: vehicle event script is running
        c: world event script is running
    $E8 ?t???ls?
        t: airship is taking off
        l: arrows are not locked
        s: arrows are shown
    $E9 ?????????
++$EA event script pointer
+$ED event script offset (high bit does something)
    $EF movement distance/pause duration
    $F0 current event command
+$F1
    $F3 character movement speed
+$F4 destination map index ???
    $F6 facing direction (0 = up, 1 = right, 2 = down, 3 = left)
    $F7 graphical action
    $FA 4 frame counter (magitek train ride)
+$FA origin X
+$FC origin Y

```

\$0100-\$FFFF: World RAM

```

$0140-$09FF mode 7 HDMA data
$0A00-$0AFF dp stack
$0B00-$0B0F

```

```

    $11F0 mosaic intensity
    $11F2
    $11F3
    +$11F4

```

```
+$11F6 ???????? ?????gbm
      g: map graphics are already loaded
      b: enable battle
      m: hide mini-map
$11F8
$11F9 battle bg index
$11FA vehicle index (0 = none, 1 = airship, 2 = chocobo)
$11FB showing character's graphic index
$11FC showing character's palette index
++$11FD world map event pointer

+$1F64 map index

+$1F60 character XY position (in tiles)
+$1F62 airship XY position (in tiles)
+$1F64 map index
+$1F66 map XY position
    $1F68 facing direction (bit 7 does something too)
+$1F69 parent map index
+$1F6B parent XY position
    $1F6D random number
+$1F6E random battle counter
$1F70-$1F7F saved character palette indexes
    $1F80 current song index
$1F81-$1FA0 saved object map indexes
    $1FA1 random number pointer for random monster battle
    $1FA2 random number pointer for monster formation
    $1FA3 random number counter for monster formation
    $1FA4 random number counter for random monster battle
    $1FA5
    +$1FA6 pointer to current showing character's object data
$1FA8-$1FBF saved counter data
    +$1FC0 party XY position
$1FC2-$1FD1 event words
    +$1FCE number of dragons left
    $1FD2 parent map facing direction
$1FD3-$1FF2 character saved XY positions (2 bytes each)
$1FF3-$1FF6 party facing directions
$1FF7-$1FFD -
    +$1FFE SRAM checksum

$6B30-$6D4F sprite data
$6D50-$6E4F
$6E50-$6ECF

$6F50-$734F map tile formation
$7350-$934F map graphics (8x8 tiles)
$9350-$93DF map tile palette assignments

$B5D0-$B64F
```

```

    $B650 walking animation position (0-3)
$B750-$B84F water tile graphics
$B850-$B85F water tile movement counters (1 per line)
    $B860

$E000-$E200 color palettes

$F120-$F7FF saved $0520-$0BFF
$F800-$FFFF decompression buffer

```

\$7F0000-\$7FFFFFFF: World Map Data

Tile data for the world map. The map is 256×256 tiles, with one byte for each tile.

Magitek Factory Train Ride

```

$7E2000-$7E9C44 tile graphics (29 items, 1097 bytes each)

$7F01E0-$7F02FF layer 16-21 tile data (6x12 items, 4 bytes each) last layer
$7F0300-$7F03BF layer 12-15 tile data (4x12 items, 4 bytes each)
$7F03C0-$7F041F layer 10-11 tile data (2x12 items, 4 bytes each)
$7F0420-$7F047F layer 8-9 tile data (2x12 items, 4 bytes each)
$7F0480-$7F04DF layer 6-7 tile data (2x12 items, 4 bytes each)
$7F04E0-$7F050F layer 5 tile data (1x12 items, 4 bytes each)
$7F0510-$7F053F layer 4 tile data (1x12 items, 4 bytes each)
$7F0540-$7F056F layer 3 tile data (1x12 items, 4 bytes each)
$7F0570-$7F059F layer 2 tile data (1x12 items, 4 bytes each)
$7F05A0-$7F05CF layer 1 tile data (1x12 items, 4 bytes each)
$7F05D0-$7F05FF layer 0 tile data (1x12 items, 4 bytes each) first layer
-----
    $00 x position
    $01 y position
    +$02 tile index (pointer to tile pointers at $0800, $0198-$0318
are $0000-$0180 flipped horizontally)

$7F0800-$7F0AB7 pointers to magitek train ride tiles, copied from $D8DD00
(29 items, 24 bytes each, +$7E0000)
$7F0AB8-
$7F0AF8-
$7F0B38-    pitch data 1
$7F0B78-    pitch data 2
$7F0BB8-    yaw data
$7F0BF8-    yaw multiplier data
$7F0C18-    background data

    +$7F0C5C pointer to tile data

$7F9618-$7FE617 magitek train ride graphics buffer (80 items, 256 bytes

```

each)

World VRAM

\$0000-\$3FFF map graphics (8 bpp)

\$4000-\$43FF bg1 map formation

\$4400-\$4BFF bg2 map formation

\$4C00-\$4FFF bg3 map formation

\$5C00-\$5FFF font graphics (2 bpp)

\$6000-\$7FFF sprite graphics (4 bpp)

From:

<https://www.ff6hacking.com/wiki/> - **ff6hacking.com** wiki

Permanent link:

https://www.ff6hacking.com/wiki/doku.php?id=ff3:ff3us:doc:asm:ram:world_ram

Last update: **2021/04/21 23:15**

