Pilot Hacking Guide

by Soren Kanzaki Updated to v0.9 on May 23, 2002

```
Super Robot Taisen A Pilot Hacking Guide v.0.9
Released on May 23, 2002
by Soren Kanzaki (soren_kanzaki@yahoo.com)

-------------------
Table of Contents:
-------------------

Section 1: Overview
Section 2: Version History
Section 3: Pilot Memory Block Basics
Section 4: Byte by Byte
Section 5: Advanced Information
  Part A: Enemy Pilot Basics
  Part B: Game Logic
Section 6: Pilot Block Address Table
Section 7: Pilot Background Data Code Table
Section 8: Seishin Code Table
Section 9: Mecha Assignment Code Table
Section 10: Moves Left Address Table
Section 11: Supplement: Memory, Binary, and Hexidecimal
Section 12: Credits
Section 13: Copyright / Authorization
Section 14: Miscellaneous

--------------------
Section 1: Overview
--------------------

    A story about giant robots without any pilots ... well, it's not much of
a story, is it?  Mecha are nice, but it's the interactions and emotions of
the young men and women (and occasionally, older men, women, robots, aliens
...) that ride these futuristic knights into battle that keeps the story
grounded in terms that everyone can understand.

    But what if you think that Fa Yuiry should be more skilled than anyone
else?  Perhaps you feel that Amuro Rey can't ever, ever get hit?  Maybe you'd
just like to recruit Anavel Gato and see how strong he is.

    It's time to get out your hacking hat, and explore the secrets of what
REALLY makes a Newtype tick ...

---------------------------
Section 2: Version History
---------------------------

    0.3 (5/03/02): First draft, a.k.a. Jaburo Edition.  Covers Universal
Century pilots.

    0.4 (5/09/02): Gundam Fight Edition.  Covers Mobile Fighter G Gundam
pilots.

    0.5 (5/14/02): Operation Meteor Edition.  Covers Shin Kidou Senki Gundam
Wing pilots.
```

0.9 (5/23/02): Wow, look at those skipped versions! Why, you ask? Oh, nothing much. I just added all the rest of the pilots and codes for all of the data in the game. This document is now complete, excepting the potential addition of a usage guide (I didn't think it was that hard to use, but apparently ...) and any name/translation corrections.

Section 3: Pilot Memory Block Basics

I'm positive you've seen this disclaimer and much of this section somewhere before:

If you use any of these cheats, I'm not responsible for any 'weird' things happening to your game or your save data. You use these cheats at your own risk (to your game, your system, your enjoyment of Super Robot Taisen A).

I made this document as a sort of educational glimpse into how the game was put together. You can make the game easier. You can make it harder. You can make it more fun. You can make it a bore. I think you get the point.

Secondly, this document is much more technical in nature than other things I have written. I cannot guarantee it's 100% correct. I cannot guarantee you'll understand it. Hopefully, both of those conditions will hold true.

Now, with the formalities over ...

Super Robot Taisen A uses a two-tiered system to store information about virtually everything in the game. One tier is the hard-coded, 'background' information written on the game itself. This includes all the pictures for all the pilots and mecha, their dialogue, their battle AI (do I flee at x% HP?), their base level statistics, how much ammunition a Beam Rifle can hold, etc. This data is, for all intents and purposes, unchangeable. Not that we need to change it, anyway.

The second tier is the data stored in the memory. A variety of devices (such as a Gameshark) allow us to directly access that memory and write all sorts of things into it. I will not go into the step-by-step detail of how to alter this data, but it is assumed that you are able to do so. It is through this method that one can customize the pilots of Super Robot Taisen A.

Like their mecha counterparts, all friendly pilots have blocks of data stored consecutively at fixed addresses in memory. Unlike their robots, the pilots only take up 32 bytes a piece. Enemy pilots are, once again, stored in dynamically allocated blocks after friendly pilots. The pilot data begins with everyone's favorite Newtype who wears a dress, Lalah Sune. She resides at 200c110.

As previously promised, we shall dwell briefly on a block of 'temporary' data. This data has but one use at present, and that is permanently allowing a pilot to keep moving. (One use, yet a powerful one!) These blocks seem to be 72 bytes long and interlace friends and enemies in an order that is probably only understandable to the programmers. In section 10, I provide the addresses of the 'Moves Left' byte only. It's sketchy, but I thought some people might enjoy this information. The Kakusei Seishin, when cast,

increments this from 1 to 2, which causes a W to appear in the Pilot's Quick Status (put the cursor over the pilot and hit B). We'll discuss how Seishin work in more detail in sections 5.B and 8.

Time to look at Lalah Sune under the microscope, as it were. Lalah's pilto data runs from 200c110 to 200c12f. This data holds at LEAST the following (there are significant gaps in the pilot data block at present):

The Current Seishin Toggles Active on this Character (more on this later)
The Pilot Background Data Code (more on this later)
The Current Mecha to which this Character is Assigned
The Current Kill Count
The Cumulative Experience of this Character
The Current Seishin Points this Character has Remaining
The Current Kiryoku of this Character
The Number of Support Actions Remaining for this Character
The Pilot Enable Byte (more on this later)

Sadly, there are rather large gaps (15 bytes, almost half!) in my knowledge of this block. Still, people kept asking, so I decided that releasing this data was in order. Reverse peeking (as described in the Mecha Guide) hints that some of this data may store the number of attacks/dodges/uses of a given statistic that has occurred to date, which may in turn alter the current statistics of the pilot. I have not yet confirmed this.

Normally, the game will initialize this data correctly (that is, look to the game cartridge to load Lalah's data in the Lalah's memory block), based on game events, and operate this data normally (if you kill something, it increments your kill count, etc. etc.). We can manipulate this data, however, and that's what this guide is about. Ready?

Section 4: Byte by Byte

Let's look at each section of the data block, then. We'll start a byte 0, and work our way onwards. Next to each number is the address of that byte for Lalah (as an example). We'll talk about where the rest of the pilots are in a later section.

Byte 0 (200c110): Seishin Toggle Byte 1
Byte 1 (200c111): Seishin Toggle Byte 2
Byte 2 (200c112): Seishin Toggle Byte 3
Byte 3 (200c113): Seishin Toggle Byte 4
Bytes 4-5 (200c114): Pilot Background Data Code
Bytes 6-9 (200c116, 200c118): Unknown
Bytes 10-11 (200c11a): Mecha Assignment
Bytes 12-13 (200c11c): Current Kill Count
Bytes 14-15 (200c11e): Cumulative Experience
Bytes 16-17 (200c120): Current Seishin Points
Byte 18 (200c122): Current Kiryoku
Byte 19 (200c123): Support Actions Remaining
Bytes 20-29 (200c124-200c12d): Unknown
Byte 30 (200c12e): Pilot Enable Byte
Byte 31 (200c12f): Unknown

See what I meant about the unknowns? Anyway:

Seishin Toggle Byte - these bytes controlled which Seishin have been 'cast'

upon the character. You can view these on the Quick Status or Seishin screens of the character. Not all are valid, and this data is ... binary! (See sections 5.B and 8 for additional details.)

Pilot Background Data Code - like it's mecha brethren, the PBDC determines the picture, base statistics, and battle logic of the given character. This is how you can turn Sayla Mass into Char Aznable.

Mecha Assignment - the current friendly mecha this pilot is controlling. The prime use for this byte is to force new pilots into their mecha, or to put pilots into mecha which they cannot normally pilot. It is best to coordinate this value with the Pilot data in the Mecha Data Block.

Current Kill Count - self-explanatory.

Cumulative Experience - this means the total amount of XP the character has gained. If the character has 1200 XP, then they are a level 3 character with 300 XP until the next level, and so on. Exceeding 50,000 (level 100) will result in VERY odd behavior.

Current Seishin Points and Kiryoku - self-explanatory.

Support Actions Remaining - self-explanatory. Support Actions remaining are shown in the Quick Status, lower-left hand corner. This included both Support Attack and Support Defense actions.

Enable Byte - Ah! This determines whether or not you 'have' this pilot. If it is on (1), the pilot will be displayed in your intermission status screens. Note that transformations and variants need not be 'on' to be usable (the Mobile Fighter G Gundam pilot's Super/Hyper modes).

Now, to modify a pilot other than Lalah, you simply add the byte number to the address in Section 6, the Pilot Block Address Table. To find out what value to put in the various bytes, look it up in the appropriate section.

Section 5: Advanced Information

Part A: Enemy Pilot Basics

Enemy pilots do not act like friendly pilots, nor do they have the same information as friendly pilots. Enemy pilots do not have any Seishin, with the possible exception of Jibaku (which, as many agree, is a lousy Seishin.) However, they have 999 Seishin points (for what that's worth ...).

Enemy pilots also have another critical piece of information that friendly pilots lack. This is their retreat threshold. While friendly pilots, being made of sterner stuff, will die before retreating, enemy pilots often have a percentile limit to the amount of damage they will stand before fleeing the battlefield. As yet, I have not figured out how to alter this information (there is a good chance it is hard-coded into the pilot, since pilots that have been replaced with the PBDC will flee even if nothing else has been altered).

Apart from that, enemy and friendly pilots are identical. (I could say something about how that's a very deep, philosophical statement, but as it has nothing to do with pilot data hacking, I will skip it.)

Part B: Game Logic

Thankfully, there are far fewer rules that must be obeyed when dealing with pilots. Primarily, this is a good place to discuss how the game deals with Seishin.

Pilots have a selection of up to 6 Seishin to cast. Now, before a Seishin can be cast, the game checks the Seishin Toggle Bytes to see if the Seishin has already been cast on this unit/pilot. Some Seishin (Tamashii, Nekketsu, Kasoku, Hirameki, Tekagen, Shuuchuu, Hicchuu, Doryoku/Ouen, Kou'un/Shukufuku, Teppeki, Kakusei, Kishuu, Totsugeki, and Kiseki) cannot be cast if they have already been cast on this unit.

Assuming the game is satisfied that this is a legal target, the appropriate Seishin Toggle is flipped. (See Section 8 and 11.) In the case of Kakusei, however, something ELSE happens. The game goes to the temporary mecha data block (Section 10). The value there must be 1 (the mecha cannot have moved) and then the game proceeds to make it 2. This way, after the mecha moves/attacks, the value goes back to 1. All the Kakusei Seishin Toggle does, then, is tell the system that Kakusei cannot be cast again on this unit! It's the internal game logic that 'implements' Kakusei by finding the correct temporary data block and adding the appropriate numbers.

Now, for the 'batch' Seishin (Kishuu, Kiseki), the toggles don't seem to work. I think the game flips the batch toggles and then flips the regular toggles as well. This is more of a theory than a fact.

The last issue is the launch status of a mecha/pilot. If the mecha does not have a pilot, then the mecha will not, I believe, show up on the launch screen. But what if it the mecha thinks it has a pilot, but the pilot does think they have a mecha? I believe the screen will allow the mecha to launch anyway. So the mecha assigned value in the pilot block is more of a courtesy piece of information. This is NOT 100% tested. You may (and I recommend you do) have to coordinate both bytes.

Section 6: Pilot Block Address Table

Remember, this is the beginning of each pilot block. To find a particular address inside this block, add the byte number to the listed address. As this are memory addresses, they are in hexadecimal, and all hexadecimal math rules apply!

200c110 - Lalah Sune
200c130 - Shiro Amada
200c150 - Aina Sahalin
200c170 - Norris Packard
200c190 - Kou Uraki
200c1b0 - South Burning
200c1d0 - Camille Bidan
200c1f0 - Quattro Bajina
200c210 - Fa Yuiry
200c230 - Four Murasame
200c250 - Rosmaia Badam
200c270 - Judau Ashta
200c290 - Roux Louka

200c2b0 - Sayla Mass
200c2d0 - Elpe Puru
200c2f0 - Puru 2
200c310 - Amuro Rey
200c330 - Bright Noah
200c350 - Kayra Su
200c370 - Domon Kasshu
200c390 - Domon Kasshu (Red/Berserk Background)
200c3b0 - Domon Kasshu (Yellow/Clear and Serene Background)
200c3d0 - Rain Mikamura
200c3f0 - Schwarz Bruder
200c410 - Sai Sici
200c430 - Sai Sici (Yellow/Super Mode Background)
200c450 - Chibodee Crocket
200c470 - Chibodee Crocket (Yellow/Super Mode Background)
200c490 - George de Sand
200c4b0 - George de Sand (Yellow/Super Mode Background)
200c4d0 - Argo Gulskii
200c4f0 - Argo Gulskii (Yellow/Super Mode Background)
200c510 - Allenby Biazury
200c530 - Fu'unsaiki
200c550 - Master Asia
200c570 - Master Asia (Yellow/Super Mode Background)
200c590 - Hiiro Yuy
200c5b0 - Duo Maxwell
200c5d0 - Trowa Barton
200c5f0 - Quatre Raberba Winner
200c610 - Zechs Merquise
200c630 - Lucrezia Noin
200c650 - Zhang Wufei
200c670 - Ken Wakaba
200c690 - Tapp Oceano
200c6b0 - Light Newman
200c6d0 - Maillot Plarto
200c6f0 - Kami Kappei
200c710 - Kamie Uchuuta
200c730 - Kamikita Keiko
200c750 - Haran Banjo
200c770 - Beautiful Tachibana
200c790 - Kabuto Kouji
200c7b0 - Yumi Sayaka
200c7d0 - Boss
200c7f0 - Tsurugi Tetsuya
200c810 - Homura Jun
200c830 - Duke Freid
200c850 - Grace Maria Freid
200c870 - Makiba Hikaru
200c890 - Kirika
200c8b0 - Rubina
200c8d0 - Nagare Ryuuma
200c8f0 - Jin Hayato
200c910 - Tomoe Musashi
200c930 - Saotome Michiru
200c950 - Jack King
200c970 - Mary King
200c990 - Saotome Miyuki
200c9b0 - Kuruma Benkei
200c9d0 - Tetsukan Oni
200c9f0 - Kochou Oni
200ca10 - Lisa

200ca30 - Aoi Hyouma
200ca50 - Naniwa Jyuuzou
200ca70 - Nishikawa Daisuke
200ca90 - Nanbara Chizuru
200cab0 - Matsuki Kosuke
200cad0 - Ichinoki Kenta
200caf0 - Ichinoki Kazuyoshi
200cb10 - Gou Kentarou
200cb30 - Gou Daijiro
200cb50 - Gou Hiroshi
200cb70 - Mine Ippei
200cb90 - Oka Megumi
200cbb0 - Ryuuzaki Kazuya
200cbd0 - Yuuzuki Kyoshirou
200cbf0 - Izumi Nana
200cc10 - Tenkawa Akito
200cc30 - Misumaru Yurika
200cc50 - Daigouji Gai
200cc70 - Subaru Ryoko
200cc90 - Amano Hikaru
200ccb0 - Maki Izumi
200ccd0 - Akatsuki Nagare
200ccf0 - Shiratori Tsukumo
200cd10 - Axel Aruma
200cd30 - Lamia Loveless

Section 7: Pilot Background Data Code Table

Remember, always enter this as 16-bit, unsigned data! Otherwise, the lo-byte ordering will be off, and the game will most likely crash very painfully. (The system will try to read data from a part of the game that doesn't have the right data.)

Pilots with the designation [CV] are conversational only. They are only provided to fill the gaps, and to show that this is how the game figures out which portrait to show when everyone is talking. Conversation pilots have no battle statistics (like support pilots), and usually their Seishin slots are filled with Jibaku. If they attack, they do 10 damage.

Last note: Don't mix this list up with the pilot list for the mecha block. They are totally different! (The mecha block lists friendly pilots first, then anything beyond that is dynamically generated based on enemies present in the current map.)

0 - Lalah Sune
1 - Char Aznable
2 - Ranba Ral
3 - Gaia
4 - Ortega
5 - Mash
6 - Shiro Amada
7 - Aina Sahalin
8 - Norris Packard
9 - Ghinias Sahalin
10 - Kou Uraki
11 - South Burning
12 - Nina Purpleton [CV]
13 - Anavel Gato

14 - Kelly Layzner
15 - Cima Garahau
16 - Karius
17 - Neuen Bitter
18 - Aguille Delaz
19 - Camille Bidan
20 - Quattro Bajina
21 - Fa Yuiry
22 - Astonage Medosso [CV]
23 - Four Murasame (friendly version)
24 - Four Murasame (angry, enemy version)
25 - Rosmaia Badam (angry, enemy version)
26 - Rosmaia Badam (friendly version)
27 - Judau Ashta
28 - Roux Louka
29 - Sayla Mass
30 - Elpe Puru
31 - Puru 2
32 - Haman Kahn
33 - Gremmi Toto
34 - Mashma Serro
35 - Chara Soon
36 - Rakan Dakaran
37 - Amuro Rey
38 - Bright Noah
39 - Kayra Su
40 - Char Aznable (maskless)
41 - Gyunei Gass
42 - Quess Paraya
43 - Rezun Schneider
44 - Domon Kasshu
45 - Domon Kasshu (Red/Berserk Background)
46 - Domon Kasshu (Yellow/Clear and Serene Background)
47 - Rain Mikamura
48 - DG Rain
49 - Schwarz Bruder
50 - Sai Sici
51 - Sai Sici (Yellow/Super Mode Background)
52 - Chibodee Crocket
53 - Chibodee Crocket (Yellow/Super Mode Background)
54 - George de Sand
55 - George de Sand (Yellow/Super Mode Background)
56 - Argo Gulskii
57 - Argo Gulskii (Yellow/Super Mode Background)
58 - Allenby Biazury
59 - Red Eyed Allenby Biazury (Berserk, Enemy)
60 - Fu'unsaiki
61 - Master Asia
62 - Master Asia (Yellow/Super Mode Background)
63 - Won Yun-fa
64 - Kyouji Kasshu (Devil)
65 - Kyouji Kasshu (normal, CV)
66 - Hiiro Yuy
67 - Duo Maxwell
68 - Trowa Barton
69 - Quatre Raberba Winner
70 - Zechs Merquise
71 - Lucrezia Noin
72 - Relena Darlian [CV]
73 - Lady Une [CV]

74 - Zhang Wufei
75 - Marlemeia Khushrenada [CV]
76 - Dekim Barton [CV]
77 - Ken Wakaba
78 - Tapp Oceano
79 - Light Newman
80 - Linda Plarto [CV]
81 - Aoi Wakaba [CV]
82 - Maillot Plarto
83 - Welner Fritz
84 - Karl Guyner
85 - Dan Kruger
86 - Min
87 - Gun Jemu
88 - Goru
89 - Ganan
90 - Jin
91 - Doruchenofu
92 - Girutooru Gensui [CV]
93 - Kami Kappei
94 - Kamie Uchuuta
95 - Kamikita Keiko
96 - Chiyokin [CV]
97 - Kami Heizaemon [CV]
98 - Giraa Za Buchaa
99 - Haran Banjo
100 - Beautiful Tachibana
101 - Sanjou Reika [CV]
102 - Garrison Tokita [CV]
103 - Koros
104 - Don Zausaa
105 - Commander Mireenu
106 - Commander Aisaa
107 - Commander Risaa
108 - Commander Tooresu
109 - Kabuto Kouji
110 - Yumi Sayaka
111 - Boss
112 - Tsurugi Tetsuya
113 - Homura Jun
114 - Kabuto Kenzou
115 - Ankoku Dai-Shogun
116 - Jigoku Dai-Gensui
117 - Umon Daisuke [CV]
118 - Duke Freid
119 - Grace Maria Freid
120 - Makiba Hikaru
121 - Kirika
122 - Rubina
123 - Gandaru Shirei
124 - Redigandaru
125 - Burakii Taichou
126 - Zuril Choukan
127 - Vega Kotei
128 - Nagare Ryuuma
129 - Jin Hayato
130 - Tomoe Musashi
131 - Saotome-hakushi (Dr. Saotome) [CV]
132 - Saotome Michiru
133 - Jack King

134 - Mary King
135 - Saotome Miyuki
136 - Kuruma Benkei
137 - Tetsukan Oni
138 - Kochou Oni
139 - Hakkotsu Oni
140 - Lisa
141 - Burai Daimyo
142 - Haidoraa Gensui
143 - Aoi Hyouma
144 - Naniwa Jyuuzou
145 - Nishikawa Daisuke
146 - Nanbara Chizuru
147 - Matsuki Kosuke
148 - Yotsuya-hakushi (Dr. Yotsuya) [CV]
149 - Ichinoki Kinta
150 - Ichinoki Kazuyoshi
151 - Roppeto [CV]
152 - Joutei Janera
153 - Soutou Warukimedesu
154 - Shogun Dangeru
155 - Gou Kenichi
156 - Gou Daijiro
157 - Gou Hiroshi
158 - Mine Ippei
159 - Oka Megumi
160 - Gou Kentarou [CV]
161 - Sakonji-hakushi (Dr. Sakonji) [CV]
162 - Prince Haineru
163 - Lee Katherine [CV]
164 - Rui Shankyaru
165 - Ryuuzaki Kazuya
166 - Yuuzuke Kyoshiro
167 - Izumi Nana
168 - Miwa Taichou [CV]
169 - Erika [CV]
170 - Rihiteru
171 - Aizamu
172 - Barubasu Shogun
173 - Raiza Shogun
174 - Oruban Dai-Gensui
175 - Tenkawa Akito
176 - Misumaru Yurika
177 - Daigouji Gai
178 - Subaru Ryoko
179 - Amano Hikaru
180 - Maki Izumi
181 - Akatsuki Nagare
182 - Hoshino Ruri [CV]
183 - Uribatake Seiya [CV]
184 - Ines Fresnage [CV]
185 - Haruka Minato [CV]
186 - Megumi Reynard [CV]
187 - Prospector [CV]
188 - Erina Kinjou Wong [CV]
189 - Shiratori Tsukumo
190 - Shiratori Yukina [CV]
191 - Tsukishin Genichirou (last name is possibly quite wrong)
192 - Takasugi Saburouta
193 - Akiyama Genpachirou

194 - Kusakabe Haruki [CV]
195 - Axel Aruma
196 - Lamia Loveless
197 - Winderu Mauzaa
198 - Lemon Burouning
199 - Omoikane [CV]
200 - Federation Soldier
201 - Mystery Person [CV]
202 - AI
203 - AI
204 - AI
205 - AI
206 - AI
207 - Mobile Doll
208 - Mechanoid Soldier
209 - Mechanoid Solider (enhanced)
210 - Vega-sei Soldier
211 - Giganos Soldier
212 - Giganos Soldier (enhanced)
213 - Zeon Soldier
214 - Zeon Enhanced Soldier (has the Kyouka Ningen Ability)
215 - Zombie Soldier
216 - Zombie Solider (enhanced)
217 - Marlemeia Soldier
218 - Mokuren Soldier
219 - Mokuren Soldier (enhanced)
220 - Shadow Mirror
221 - Shadow Mirror (enhanced)
222 - Bamu-sei Soldier
223 - Bamu-sei Soldier (enhanced)
224 - AI
225 - Mystery Man
226 - AI
227 - Duke Freid (not the usual one in your party, I imagine)

Section 8: Seishin Code Table

Okay, this section introduces a different type of data system, which I call the Compressed Bit Switch. What this means is, the computer looks at these bytes not as bytes, but as 8 bits in a row. Each bit has a function, and can be on or off.

What does this mean in practical terms? This means you must compute the value you want on the fly! (Since you can't enter cheats in binary for the most part.) As Section 11 goes into the math behind all this, I won't go into too much depth here. Sufficed to say:

To calculate the value to place in a Seishin Toggle Byte, look up the 'to-add' values given. Let's look at Seishin Toggle Byte 1. If we want our pilot to have Kasoku and Hirameki on them, we need to put 4+8 or 12 into the correct byte. If we wanted Shuuchuu and Hirameki, we'd put 40. See how that works? People who understand binary should see why these values are the way they are immediately.

Some toggles do nothing. They are noted as [does nothing], and are there for completeness (The game basically put all the Seishin one after another, in the order they appear on the Seishin Available screen.) Toggle them, and all it does is clutter your Seishin status bar in the Quick Status screen and

Seishin screen. For information on Kakusei and why it doesn't work, check Section 5.B.

Seishin Toggle Byte 1:
+1 = Jibaku [does nothing]
+2 = Teisatsu [does nothing]
+4 = Kasoku
+8 = Hirameki
+16 = Tekagen
+32 = Shuuchuu
+64 = Konjou [does nothing]
+128 = Hicchuu

'Best' Combination: 172 (activates everything but Tekagen)

Seishin Toggle Byte 2:
+1 = Doryoku
+2 = Shinrai [does nothing]
+4 = Teppeki
+8 = Ouen [does nothing, use Doryoku]
+16 = Dokonjou [does nothing]
+32 = Nekketsu
+64 = Kiai [does nothing]
+128 = Kou'un

'Best' Combination: 165 (activates Kou'un, Nekketsu, Teppeki, and Doryoku)
'Best' Combination if you use Tamashii: 133

Seishin Toggle Byte 3:
+1 = Datsuryoku [does nothing]
+2 = Kakusei [does nothing, see Section 5.B]
+4 = Shukufuku [does nothing, use Kou'un]
+8 = Kakuran [does nothing]
+16 = Kishuu [does nothing]
+32 = Tamashii
+64 = Hokyuu [does nothing]
+128 = Gekirei [does nothing]

'Best' Combination: 32

Seishin Application Byte 4:
+1 = Ai [does nothing]
+2 = Saidou [does nothing]
+4 = Kenshin [does nothing]
+8 = Totsugeki
+16 = Fukkatsu [does nothing]
+32 = Kiseki [does nothing]
+64 = ?? (not used, I think)
+128 = ?? (not used, I think)

'Best' Combination: 8

Section 9: Mecha Assignment Code Table

It is always better to cross-coordinate this data with that in the Mecha Data Block.

Last note: Don't mix this list up with the mecha list for the mecha

block. They are totally different! (The mecha block lists all the mecha; this lists only the normal, friendly mecha you can assign pilots to in the game. Those 151, you might say.)

0 - Gundam
1 - G Fighter
2 - G Armor
3 - G Bull
4 - G Sky
5 - Gundam (MA)
6 - Full Armor Gundam
7 - Zakrello
8 - Elmeth
9 - Char's Personal Use Gelgoog
10 - Gundam Ez-08
11 - High Mobility Zaku
12 - Apsaras
13 - Gouf Custom
14 - Gundam Test Model Number 1 Machine
15 - Gundam Test Model Number 1 Machine FB
16 - Gundam Test Model Number 3 Machine
17 - Gundam Stamen
18 - GM Custom
19 - Z Gundam
20 - Waverider
21 - Gundam Mk. II
22 - Super Gundam
23 - G Flyer
24 - Hyakku Shiki
25 - Methuss
26 - Methuss (MA)
27 - Argama
28 - ZZ Gundam
29 - G Fortress
30 - Full Armor ZZ Gundam
31 - Near Argama
32 - Qubeley Mk. II (purple)
33 - Qubeley Mk. II (brown)
34 - Nu Gundam
35 - Re-GZ (BWS)
36 - Re-GZ
37 - Ra Kalium
38 - Sazabi
39 - Shining Gundam
40 - Shining Gundam S
41 - God Gundam
42 - God Gundam H
43 - Gundam Maxter
44 - Gundam Rose
45 - Dragon Gundam
46 - Bolt Gundam
47 - Rising Gundam
48 - Gundam Spiegel
49 - Nobel Gundam
50 - God Gundam (Fu'unsaiki)
51 - Fu'unsaiki
52 - Master Gundam
53 - Master Gundam (Fu'unsaiki)
54 - Wing Zero Custom
55 - Deathscythe Hell Custom

56 - Heavyarms Custom
57 - Sandrock Custom
58 - Altron Custom
59 - Tallgeese III
60 - Taurus
61 - Taurus (MA)
62 - Dragonar Type-1
63 - Dragonar Type-2
64 - Dragonar Type-3
65 - Cavalier Type-0
66 - Dragonar Type-1 (L)
67 - Dragonar Type-2 (L)
68 - Dragonar Type-3 (L)
69 - Dragonar Type-1 Custom
70 - Dragonar Type-2 Custom
71 - Dragoon
72 - Falgen
73 - Zambot 3
74 - Daitarn 3
75 - Dai-Fighter
76 - Dai-Tank
77 - Mazinger-Z
78 - Diana A
79 - Boss Borot
80 - Minerva X
81 - Great Mazinger
82 - Venus A
83 - Mass Produced Great Mazinger
84 - Grandizer
85 - Spazer
86 - Grandizer (WS)
87 - Grandizer (MS)
88 - Grandizer (DS)
89 - Double Spazer
90 - Marine Spazer
91 - Drill Spazer
92 - Getta-1
93 - Getta-2
94 - Getta-3
95 - Texas Mack
96 - Getta Q
97 - Getta Dragon
98 - Getta Liger
99 - Getta Poseidon
100 - Mecha Tetsukan Oni
101 - Mecha Kochou Oni
102 - Shin Getta-1
103 - Shin Getta-2
104 - Shin Getta-3
105 - Combattler V
106 - Kerot
107 - Kerot (Kon V)
108 - Voltes V
109 - Daimos
110 - Galba FX II
111 - Aestivalis (Aerial) Akito
112 - Aestivalis (0G) Akito
113 - Aestivalis (Artillery) Akito
114 - Aestivalis (Aerial) Gai
115 - Aestivalis (0G) Gai

116 - Aestivalis (Artillery) Gai
117 - Aestivalis (Aerial) Ryoko
118 - Aestivalis (0G) Ryoko
119 - Aestivalis (Artillery) Ryoko
120 - Aestivalis (Aerial) Hikaru
121 - Aestivalis (0G) Hikaru
122 - Aestivalis (Artillery) Hikaru
123 - Aestivalis (Aerial) Izumi
124 - Aestivalis (0G) Izumi
125 - Aestivalis (Artillery) Izumi
126 - Aestivalis (Aerial) Akatsuki
127 - Aestivalis (0G) Akatsuki
128 - Aestivalis (Artillery) Akatsuki
129 - Nadesico
130 - Nadesico (Y-Unit)
131 - Aestivalis (Lunar Surface Frame)
132 - Soul Gain
133 - Vysaga
134 - Angelgu
135 - Ash Saber
136 - Raaza Angurifu
137 - God Gundam H (Fu'unsaiki)
138 - Dragonar Type-3 Custom
139 - Dai-Tetsujin
140 - Gundam Maxter S
141 - Gundam Rose S
142 - Dragon Gundam S
143 - Bolt Gundam S
144 - Master Gundam S
145 - Master GS (Fu'unsaiki, left facing)
146 - Getta-1 [non-transforming]
147 - Getta Dragon [non-transforming]
148 - God Gundam H (Fu'unsaiki) [more powerful variant]
149 - Master GS (Fu'unsaiki, right facing) [more powerful variant]
150 - Gundam Maxter S [possibly the boxing variant]

Section 10: Moves Left Address Table

A few words. There is no guarantee what number a mecha will be - this is dependent on the order in which the game places the mecha on the map ... sort of. This could be enemy mecha (which will cause the game to go into an infinite loop!) or friendly mecha. Be forewarned! Basically, though, the 'Mech Deployed' information is correct.

Each of these values is one byte (8-bit). Normally it is 1 if the mecha hasn't moved, 0 if it has, and 2 if you use Kakusei. You'll know that a unit has received this benefit if the letter W appears in the Seishin Points section of the Quick Status. (W is a short-hand in Japanese for Double.)

One more thing. If you use too many of these codes (28 didn't do it, but somewhere around 30 did), the game is liable to take a header off the deep end. (An archaic way of saying that it's going to crash, and it's going to crash hard.)

2005ca3 - Moves Left, Mech Deployed 1
2005ceb - Moves Left, Mech Deployed 2
2005d33 - Moves Left, Mech Deployed 3
2005d7b - Moves Left, Mech Deployed 4

2005dc3 - Moves Left, Mech Deployed 5
2005e0b - Moves Left, Mech Deployed 6
2005e53 - Moves Left, Mech Deployed 7
2005e9b - Moves Left, Mech Deployed 8
2005ee3 - Moves Left, Mech Deployed 9
2005f2b - Moves Left, Mech Deployed 10
2005f73 - Moves Left, Mech Deployed 11
2005fbb - Moves Left, Mech Deployed 12
2006003 - Moves Left, Mech Deployed 13
200604b - Moves Left, Mech Deployed 14
2006093 - Moves Left, Mech Deployed 15
20060db - Moves Left, Mech Deployed 16
2006123 - Moves Left, Mech Deployed 17
200616b - Moves Left, Mech Deployed 18
20061b3 - Moves Left, Mech Deployed 19
20061fb - Moves Left, Mech Deployed 20
2006243 - Moves Left, Mech Deployed 21
200628b - Moves Left, Mech Deployed 22
20062d3 - Moves Left, Mech Deployed 23
200631b - Moves Left, Mech Deployed 24
2006363 - Moves Left, Mech Deployed 25
20063ab - Moves Left, Mech Deployed 26
20063f3 - Moves Left, Mech Deployed 27
200644b - Moves Left, Mech Deployed 28

Section 11: Supplement: Memory, Binary, and Hexidecimal

Okay, a few words before we begin. Why am I including this, and why here? This is a little guide to memory, binary, hexidecimal, and how and why hacking codes work. People ask me a lot of questions about the basics, and I figure that since this guide has almost all of the different type of codes in one place, it is an excellent place to put it.

This information is probably covered in a much more professional manner in your local mathematics textbook and computer science course. Of course, since a lot of gameplayers are in high-school, they may not have a decent computer science department. Not to fret.

Time to turn things over to our professor emeritus, Professor ... err ... Washuu-chan. Just Washuu-chan.

Let's start by talking about numbers. Most of us are familiar with this subject, but let's review a little, shall we? Numbers are made up of numerals (the representations of numbers, which in the English alphabet are written as 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. We could have just as well used different smiley faces, pictures of trees, or letters.) While there is no problem when writing a number from 0 to 9, what happens when we want to write a 10? Well, as you can see, we use the concept of digits. The '1' in the number '10' is in the tens place. In the number '42,375', the number 2 is in the thousands place, and so on.

You of course know this, otherwise ... well, you know all this. But what we really mean when we write '42,375' is:

"I want the number whose value equals 4 X 10,000 + 2 X 1,000 + 3 x 100 + 7 x 10 + 5 x 1." 10 is, of course, 10 to the first power; 100 is 10 to the second power; 1000 is 10 to the third power, and so on and so forth.

This system of representing numbers is known as decimal. (From the Greek for 10.) But what if we don't want to use decimal?

To a computer, decimal is far to hard and inefficient. Computers understand two basic things: On (electrons flowing) and Off (electrons not flowing). If On = 1 and Off = 0, we can still represent any number we like. This is known as binary. So, we can write '27' in binary as 11011. (That's 1 x 16 + 1 x 8 + 0 x 4 + 1 x 2 + 1 x 1.) 16 is 2 to fourth power, 8 is 2 to the third power, etc. etc.

Hopefully, you understand binary now. Hexidecimal is similar, except instead of using 10 or 2 as a base, we use 16. In order to represent 10, 11, 12, 13, 14, and 15 in a single symbol, we use A, B, C, D, E, and F respectively. (16 in hexidecimal is, of course, 10.) So 100 in decimal is 6 x 16 + 4 or 64 in hexidecimal.

Why even bother with hexidecimal? Ahh ... well, historically, we group 8 binary switches into a single unit known as a byte. (10011011, for example.) Therefore, a byte can store values from 255 (1111111) to 0 (00000000). It so happens that 255 is equivalent to FF in hexidecimal. See where we're going with this?

A two-digit hexidecimal value, then, can represent all the values that a byte can. This is why we use hexidecimal extensively in computers.

Okay, so now we know about hexidecimal and why we use it in computers. Memory addresses are written in hexidecimal. Why? Well, a computer needs to be able to manipulate that address and store it. As a number. How does a computer store numbers? (If you said in binary, you are right.) What's the way we write binary numbers? (If you said in hexidecimal, you are again right.)

Now, we come to the interesting part. How is data stored in memory? (Not physically, but conceptually.) Well, we store all sorts of things in a game's memory. If we want to store a number, we usually store it in a single byte (if it runs from 0-255), or 2 bytes (0-65535) or 4 bytes (basically anything bigger, up to 4,294,967,295.) This is different if you want to store potentially negative values (and I won't get into it, we rarely run into negative numbers in game memory.) Things like HP, EN, how many lives you have left, the number of shots in that rifle, etc. are usually stored in this manner.

We could also store it, instead of in true hexidecimal, as though it were decimal (since hexidecimal includes 0-9. I believe this is referred to as Binary Coded Decimal.) This tends to waste space (you ignore the power of A-F), but sometimes this is done in games. Not so much nowadays, though. So, if I stored 142 in 2-bytes, in the 'value' I'm really storing is 1 in the first of those two bytes, and the decimal value of 66 in the second. (I omit a discussion of byte-ordering until another time, simply because there are two different ways to store a multi-byte number. Don't worry about it for the present.)

We can also stored a value that corresponds to something else (like a pilot's background data.) If information is kept like this, the game knows (it's built-in) what, say, a 12 stands for. We have to plug in values to figure out what each value stands for and make long lists. Sometimes, in cases like these, a 0 is empty (doesn't correspond to something). Sometimes (like in Super Robot Taisen A) it does stand for something (Pilot 0 is Lalah Sune.)

Finally, we can use memory in a BINARY fashion. Remember: FF is REALLY 11111111. Or 8 little switches in a row than can be ON or OFF. So if we want to look at things that can be on or off (like the Seishin Toggle), we can cram 8 of those things into one byte, instead of using 8 bytes. The 'Enable Bytes' are variations on this. Basically, a 0 stands for Off, and anything else is On. (The reason for this lies within the realm of assembly language, and will not be covered here. Sufficed to say, the game has a way of quickly checking if something is on or off, and only 0 stands for off.)

Well, this ends the brief supplement of Binary, Hexidecimal, and how memory is used in most games. Hopefully, this answers a few questions.

Section 12: Credits

There are several people without whose publicly available resources this document could have never been complied:

GameFAQs (www.gamefaqs.com), for being the comprehensive game information site;

The people on the Super Robot Taisen A board at GameFAQs, for confirmations of some material and a few helpful hints here and there with later appearing pilots;

badkarma.net, whose information helped me confirm the translations of some mecha and pilot names;

Jeffrey's J<->E Dictionary Server (linear.mv.com/), an excellent on-line dictionary.

Section 13: Copyright / Authorization

Section 14: Miscellaneous

Enemy Pilots:
As you may expect, the only way to get enemy pilots is to overwrite your own. Sayla Mass and Beauty (Beautiful Tachibana) are prime candidates, since they aren't available for most of the game. You could, of course, overwrite anyone and everyone who strikes your fancy.

Edition Names:
Of course, Anaheim and Zeonic don't really have anything to do with the pilots. But Jaburo, the headquarters of the Federal Forces, seemed like an appropriate title for information on the pilots themselves.

The Gundam Fight is the series of combats that are the main point of the Mobile Fighter G Gundam Series.

Operation Meteor is the name of the operation to conquer the Earth (under the auspices of the Barton family) following the dropping of a colony onto the planet. (Sound vaguely familiar?)

Next?:
If only the Mecha Hacking Guide were as easy ^^ The problem with that is a few hard to translate weapons systems, and the fact that there's a lot of weapons that I cannot link to a mecha (probably because I'm on Scenario 14. It MIGHT have something to do with it.) I'll see what kind of headway I can make there.

Primarily I made this 'full disclosure' release for everyone who's been waiting for the data. (You know who you are. Yes, you. Over there.)