Even More Tamagotchis Were Harmed in the Making of this Presentation

Natalie Silvanovich
@natashenka

About Me

- Security Researcher at BlackBerry
  – (But I don’t represent them)
- Studied electrical engineering, but mostly into software hacking
- First-time hardware hacker/reverse engineer
- Tamagotchi enthusiast

What are Tamagotchis?

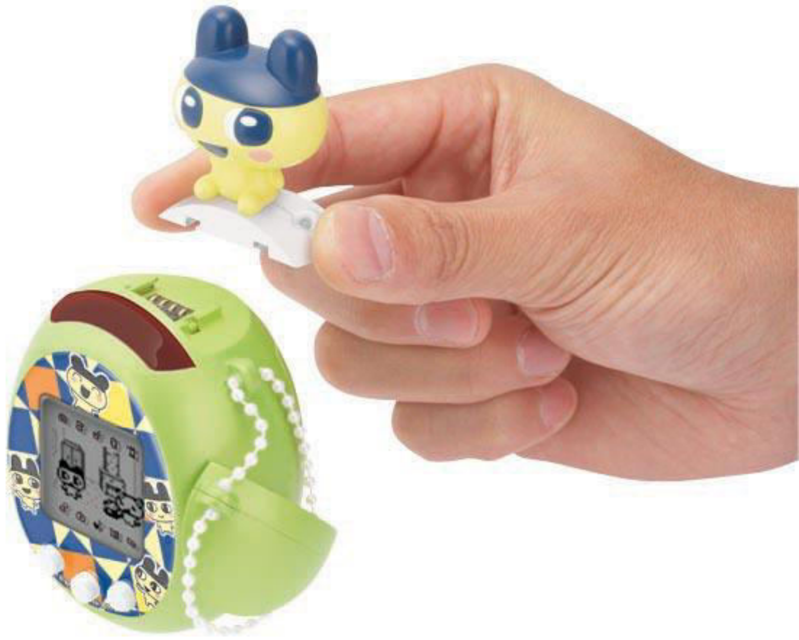
- The same virtual pet toys you remember from the 90’s
- Functionality has evolved substantially
  - Now they can go to school, have jobs, make friends, get married and have kids!
- Newer versions have an IR interface

so that they can communicate with

other Tamagotchis

TamaTown Tama-Go

- The “Christmas” Tamagotchi from 2010
- Same functionality for smaller hands
- Supports detachable ‘figures’ with extra games and stores



Goals

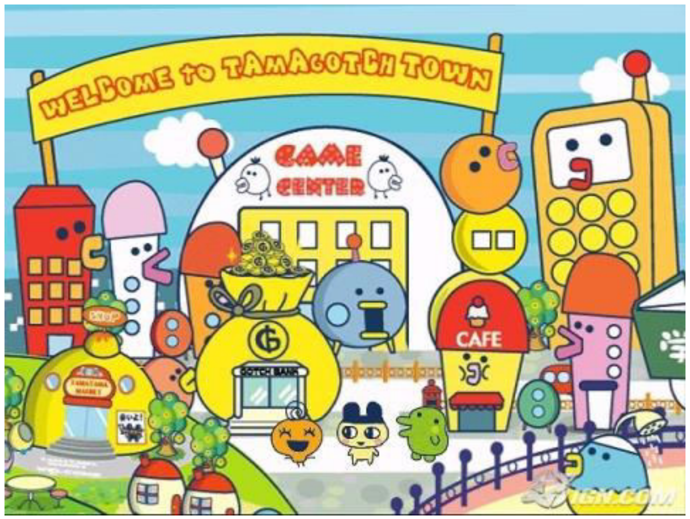
- Dump Tamagotchi code
- Answer the ‘deeper questions’ of Tamagotchi life
- Make my gotchis rich and happy
- Make a Tamagotchi development environment
- Have fun!

Previous Work

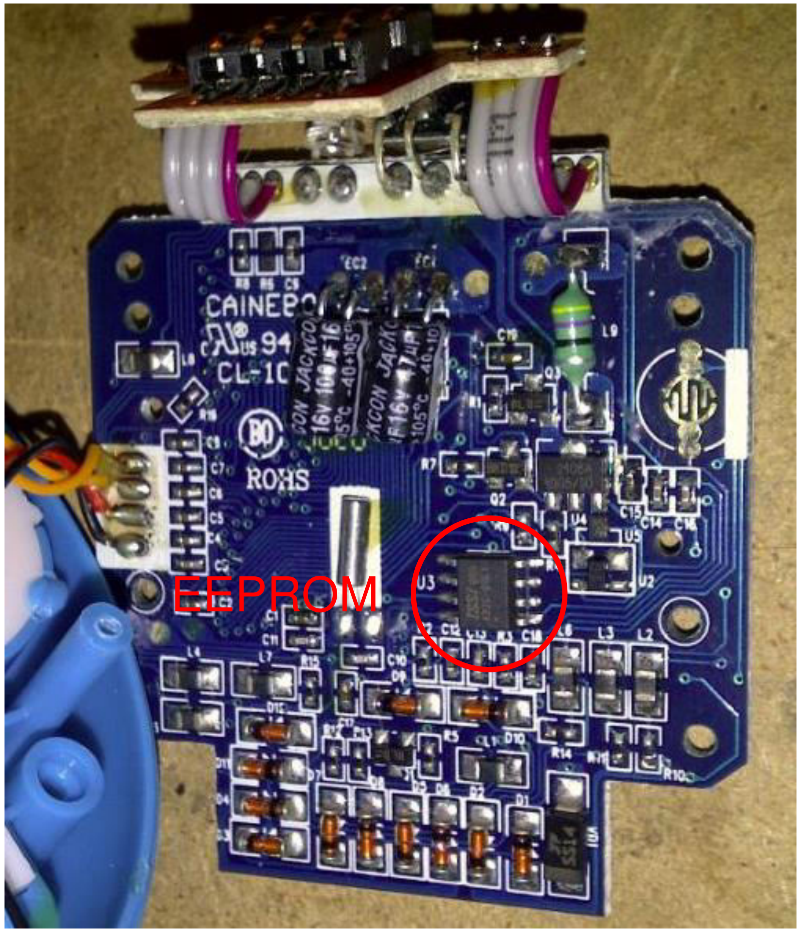
Teardown

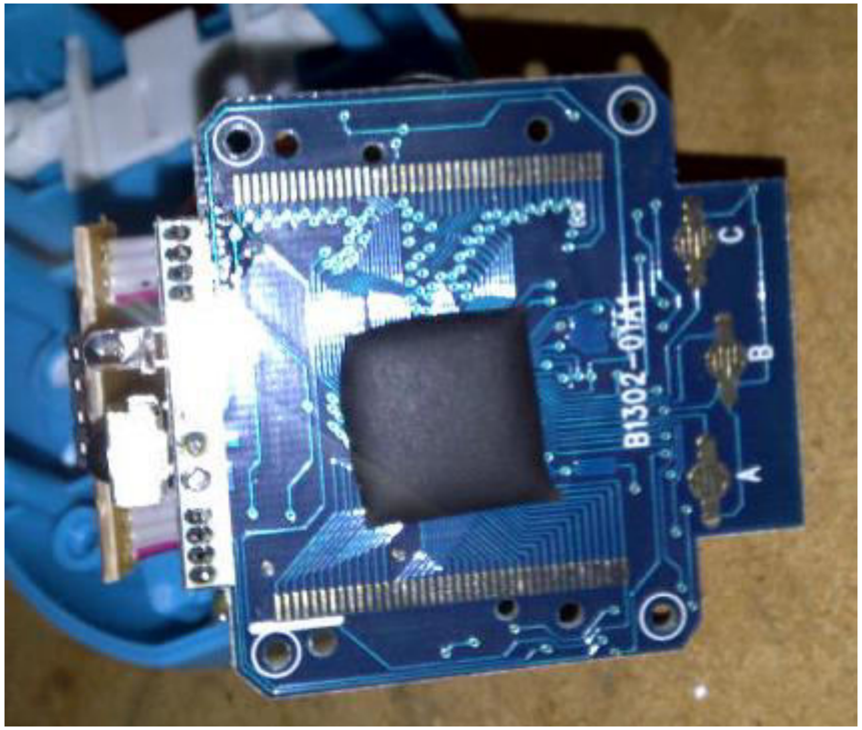
Hardware Teardown

- Took apart a Tama-Go and Tamagotchi to determine if code dumping was a possibility
- Looked for helpful interfaces
- Also took apart a figure

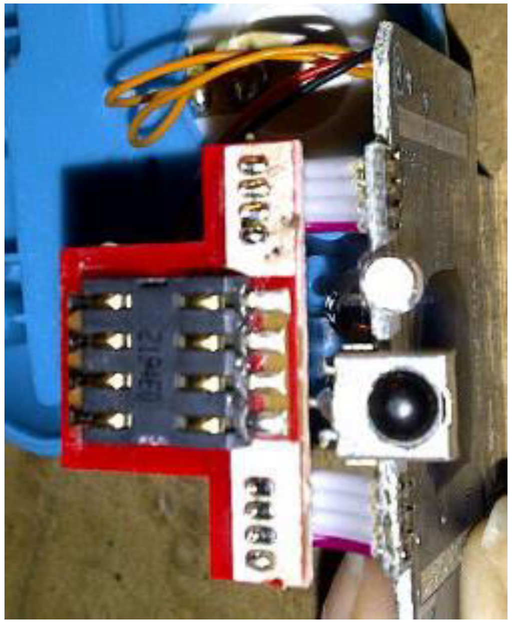
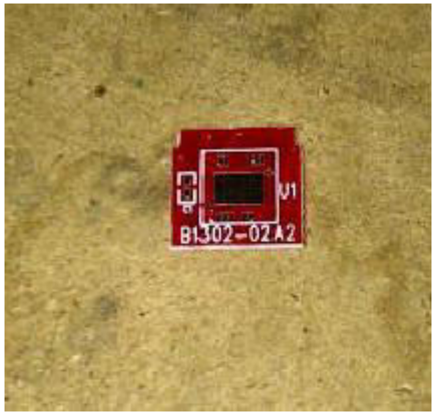


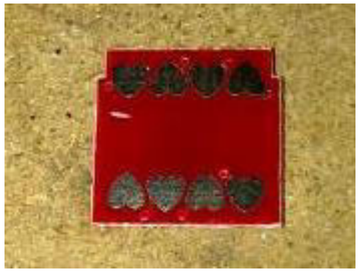
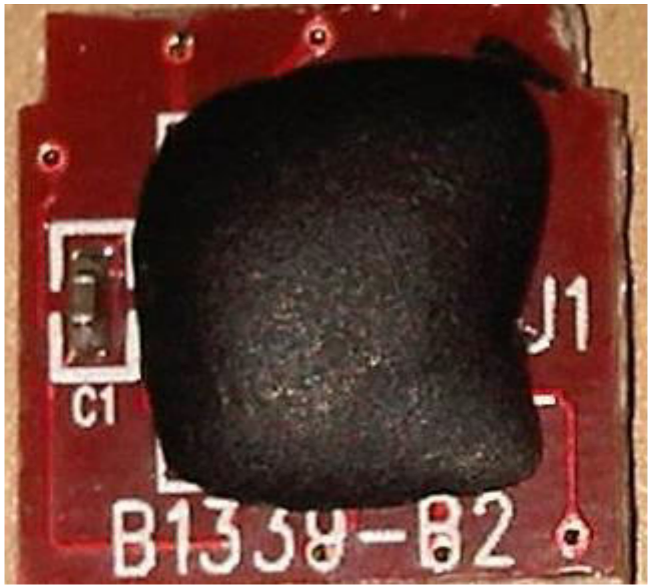
Tama-Go Board





Tama-Go Figure

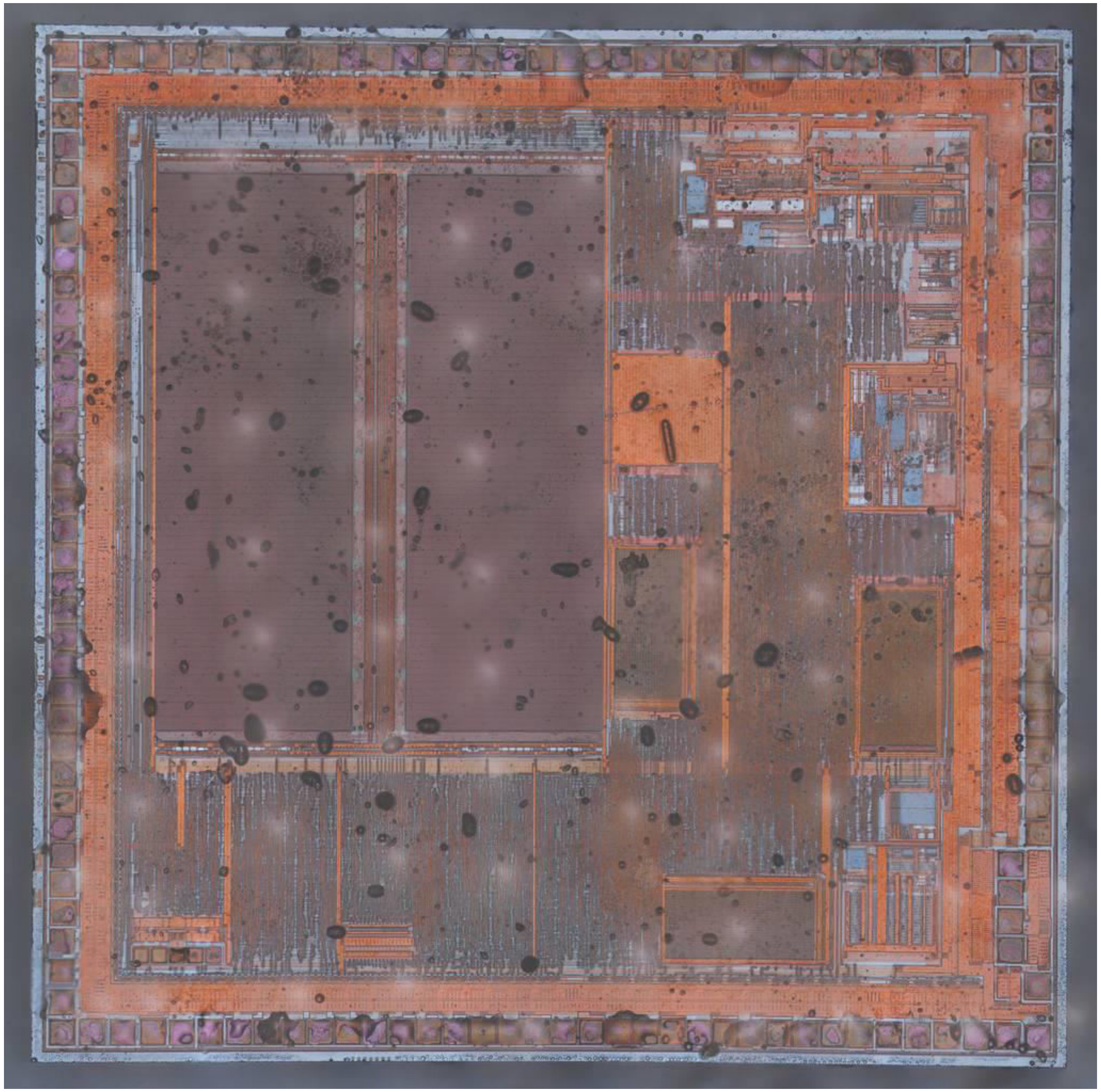



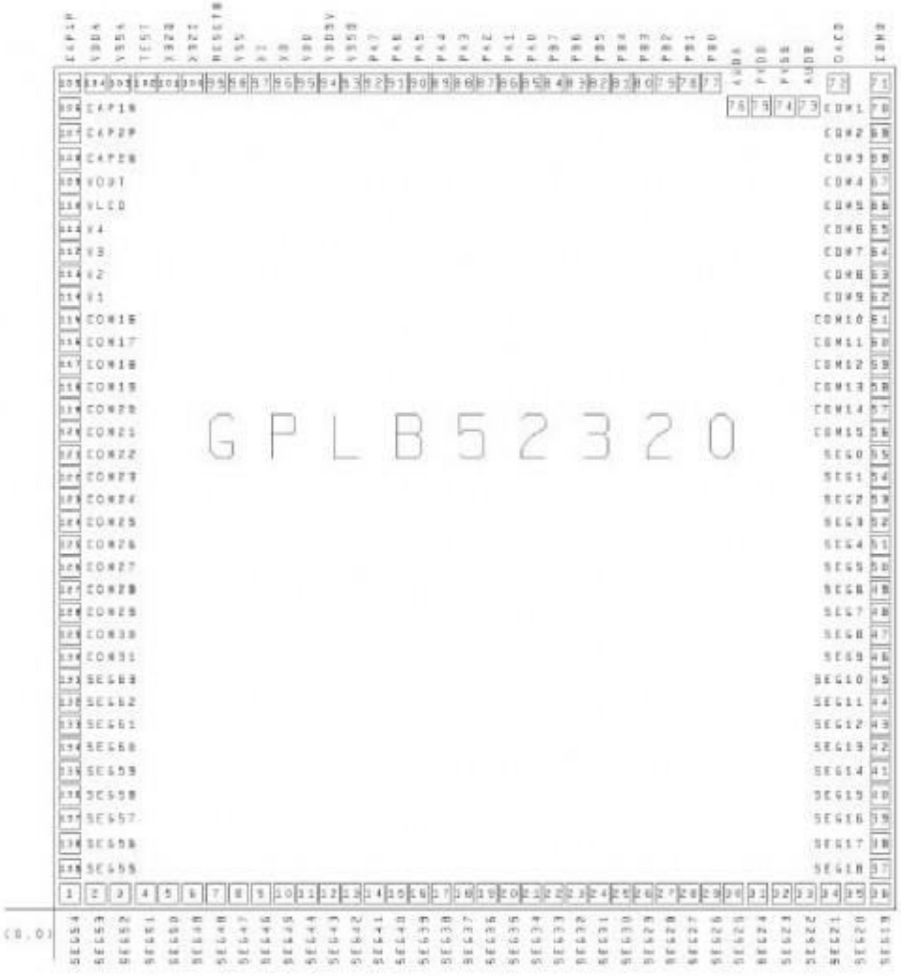
Microcontroller Identification

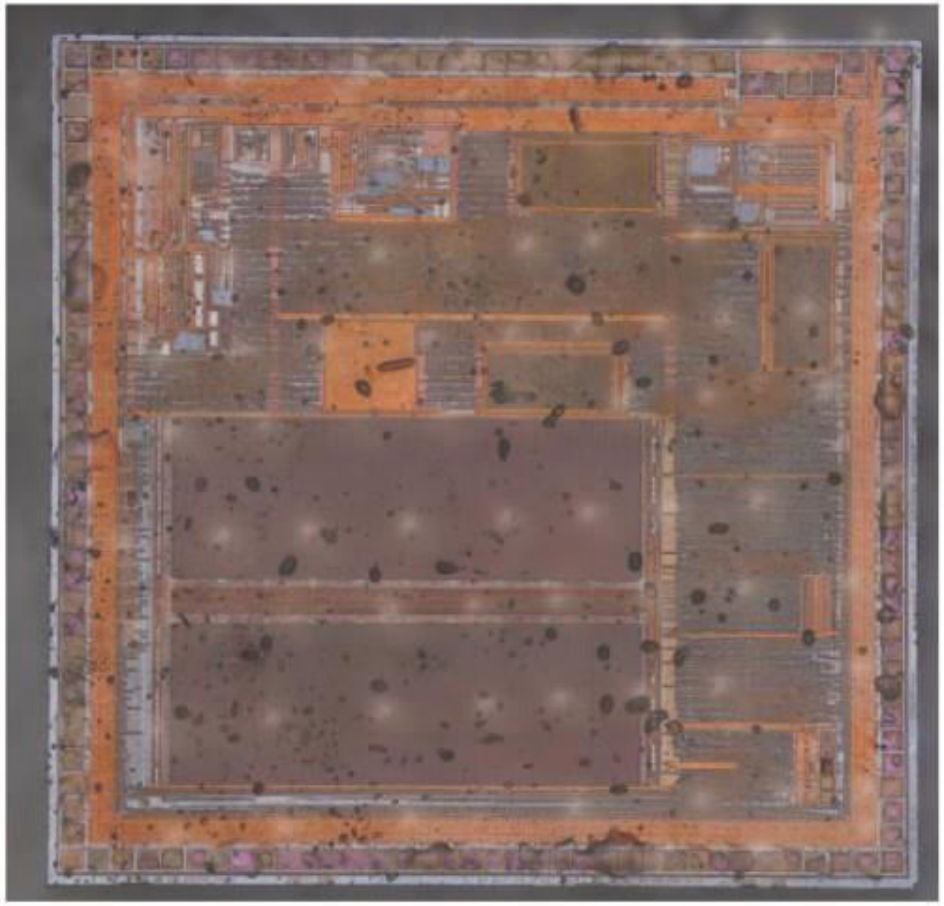
Identifying the Microcontroller

- Considering the lack of external hardware, MCU was likely under the ‘blob’
- Tried several methods to remove, including acetone, heat, a razor blade and a chopstick
- Travis Goodspeed kindly offered to decap the chip with acid



- Eventually, success!





GPLB5X Series LCD Controller

- 8 bit 6502 microprocessor
- 1536 bytes RAM
- 320 or 640 kbyte mask ROM (depending on model), baked to perfection for each customer
- 512 bytes LCD RAM
- 4 color grayscale LCD controller
- SPI
- Audio DAC

Generalplus

Dumping Mask ROM

- Not sure how to dump mask ROM, but had a few ideas
  - Restore a bad state from EEPROM
  - Look for test functionality
  - Exploit a vulnerability in figure or IR processing
  - Read ROM with a microscope
  - Pin manipulation

Test Program

Test Program?

- GeneralPlus mask ROMs contain a GP test program that can probably dump code
- Contacted GeneralPlus for a devkit
  - Requires an NDA
- Looked around online
  - No one seems to have a devkit or know the test program

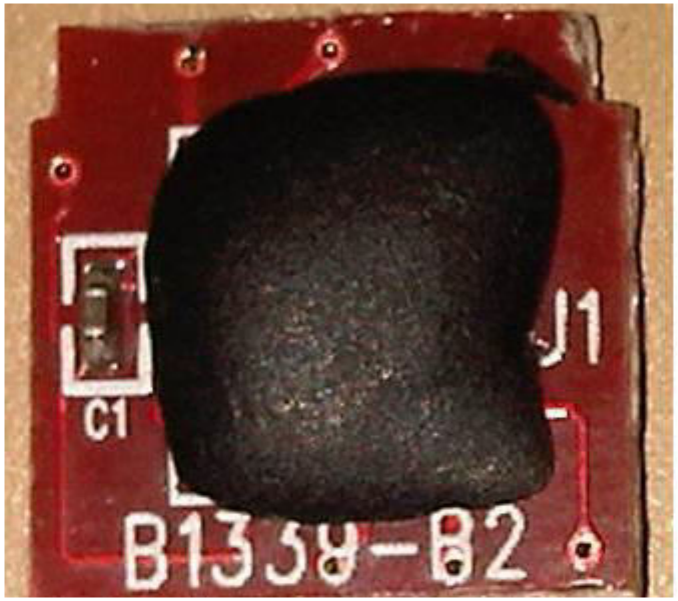
Figure ROM

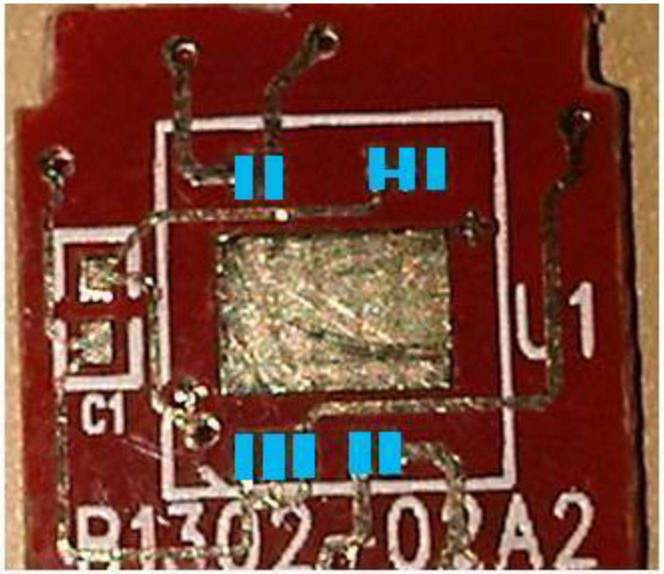
Figure ROM

- Decoding the figure ROM could be useful in a few ways
  - Making your own Tamagotchi games
  - Executing code on the Tamagotchi
  - Dumping mask ROM
  - Understanding Tamagotchi behaviour

Figure ROM Pads

- The unpopulated PCBs in lite figures appear to be the same boards used in regular figures

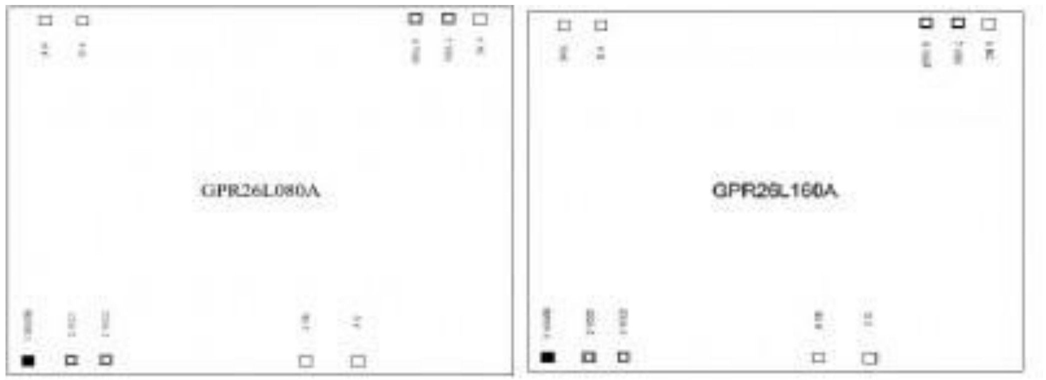




- Makes the mask ROM pad layout visible

Figure ROM Chip

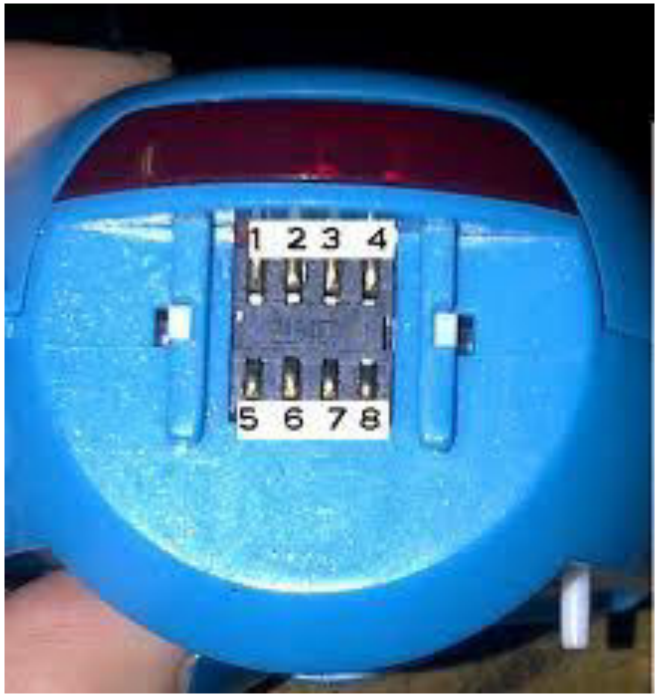
- GeneralPlus makes an SPI ROM with a similar layout



- Assumed figures use this ROM

Figure ROM Pins

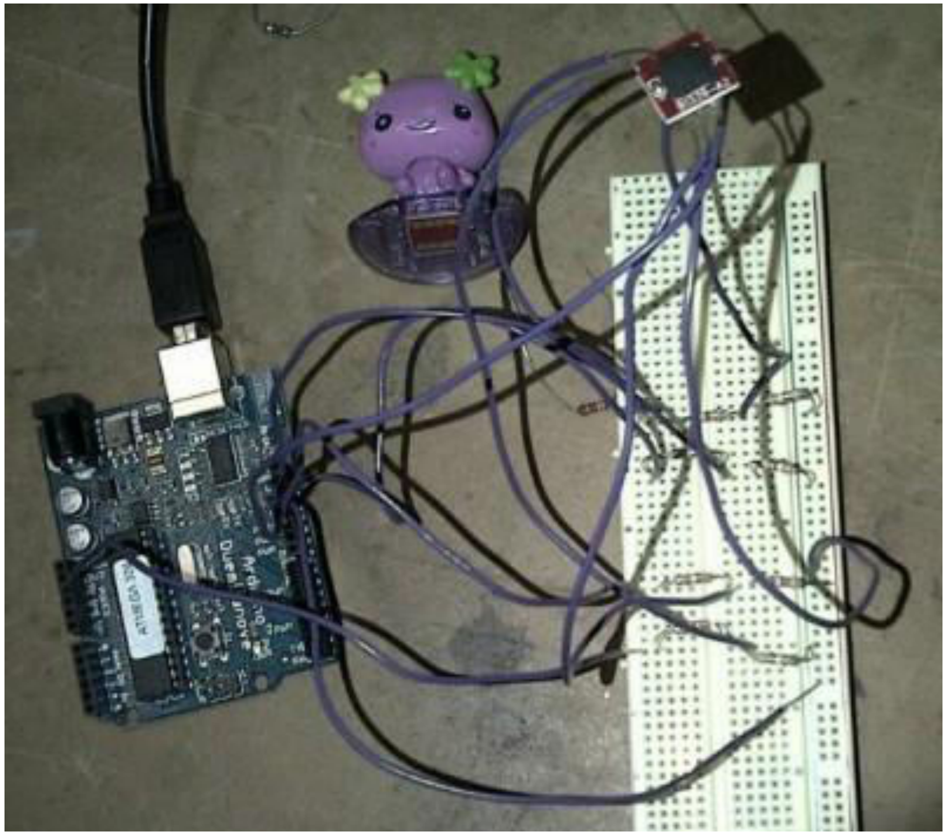
- Based on the GeneralPlus ROM datasheet, was able to identify the figure pins



1, 4 and 8: Ground/Jumper
2: Serial clock (C)
3: Serial data input (D)
5: Power
6: Chip Select (SB)
7: Serial Data Output (Q)

ROM Dump

- Dumped the ROM using an Arduino as SPI master

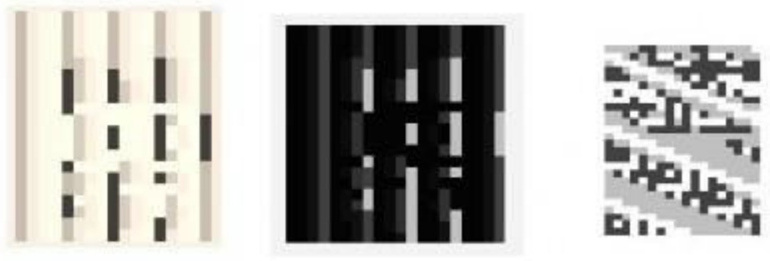


Decoding ROM

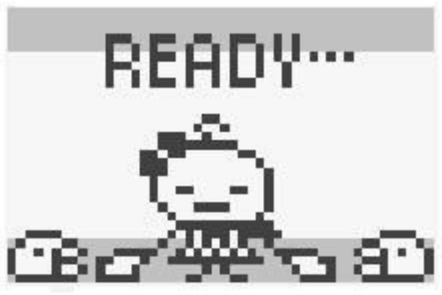
- The Tamagotchi has a four-tone display, so looked for strings of 0x00, 0x55, 0xAA and 0xFF, representing images
- Noticed that these strings were preceded by values which were reasonable for length and width

Decoding Images

- Tried decoding these images

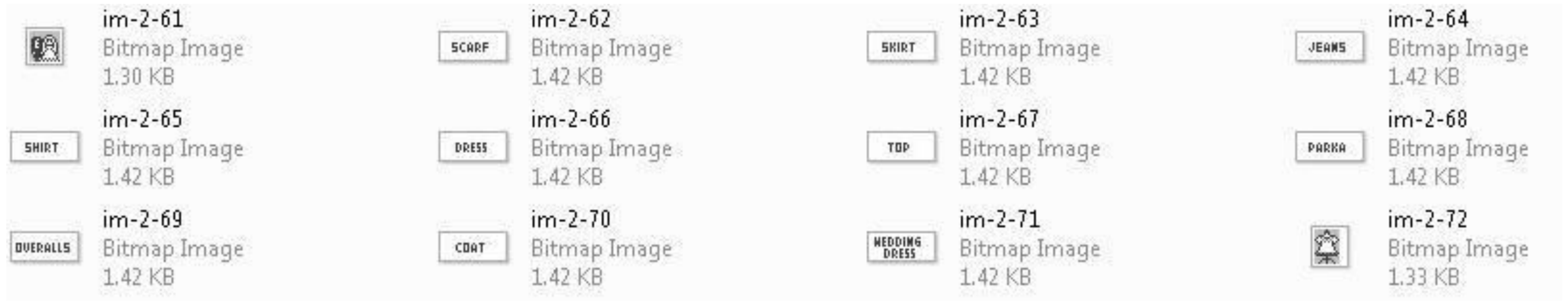


- Eventually, it worked!



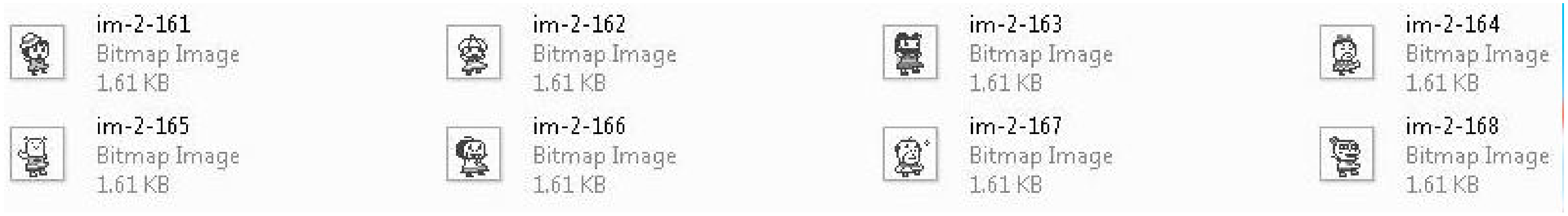
Images

- The figure contained a lot of images
- Text displays appear to be images



im-2-61 Bitmap Image 1.30 KB
im-2-62 Bitmap Image 1.42 KB
im-2-63 Bitmap Image 1.42 KB
im-2-64 Bitmap Image 1.42 KB
im-2-65 Bitmap Image 1.42 KB
im-2-66 Bitmap Image 1.42 KB
im-2-67 Bitmap Image 1.42 KB
im-2-68 Bitmap Image 1.42 KB
im-2-69 Bitmap Image 1.42 KB
im-2-70 Bitmap Image 1.42 KB
im-2-71 Bitmap Image 1.42 KB
im-2-72 Bitmap Image 1.33 KB

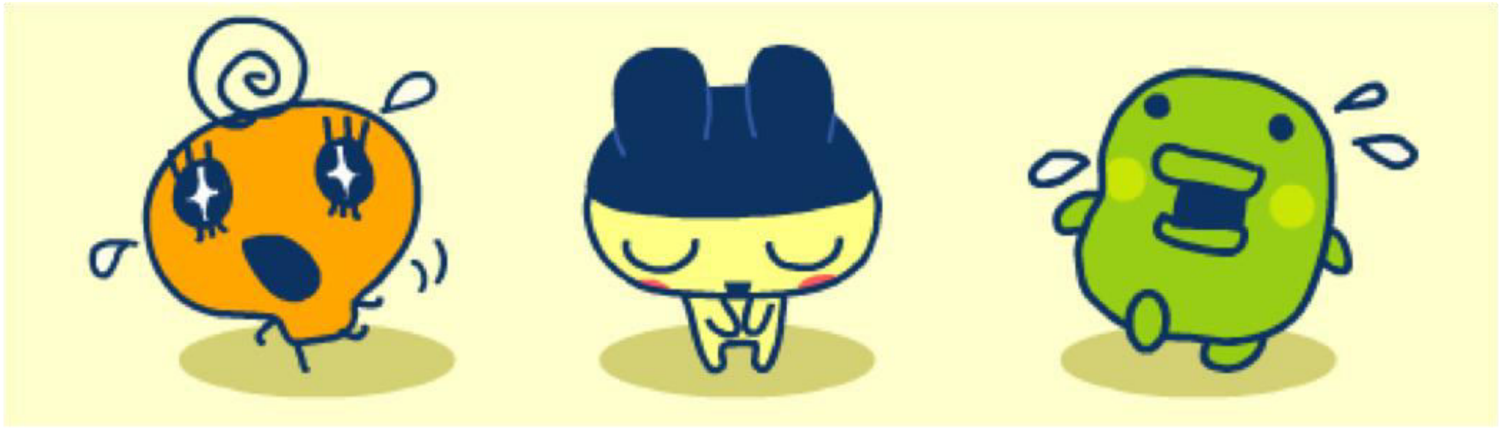
- Animations are series of images



im-2-161 Bitmap Image 1.61 KB
im-2-162 Bitmap Image 1.61 KB
im-2-163 Bitmap Image 1.61 KB
im-2-164 Bitmap Image 1.61 KB
im-2-165 Bitmap Image 1.61 KB
im-2-166 Bitmap Image 1.61 KB
im-2-167 Bitmap Image 1.61 KB
im-2-168 Bitmap Image 1.61 KB

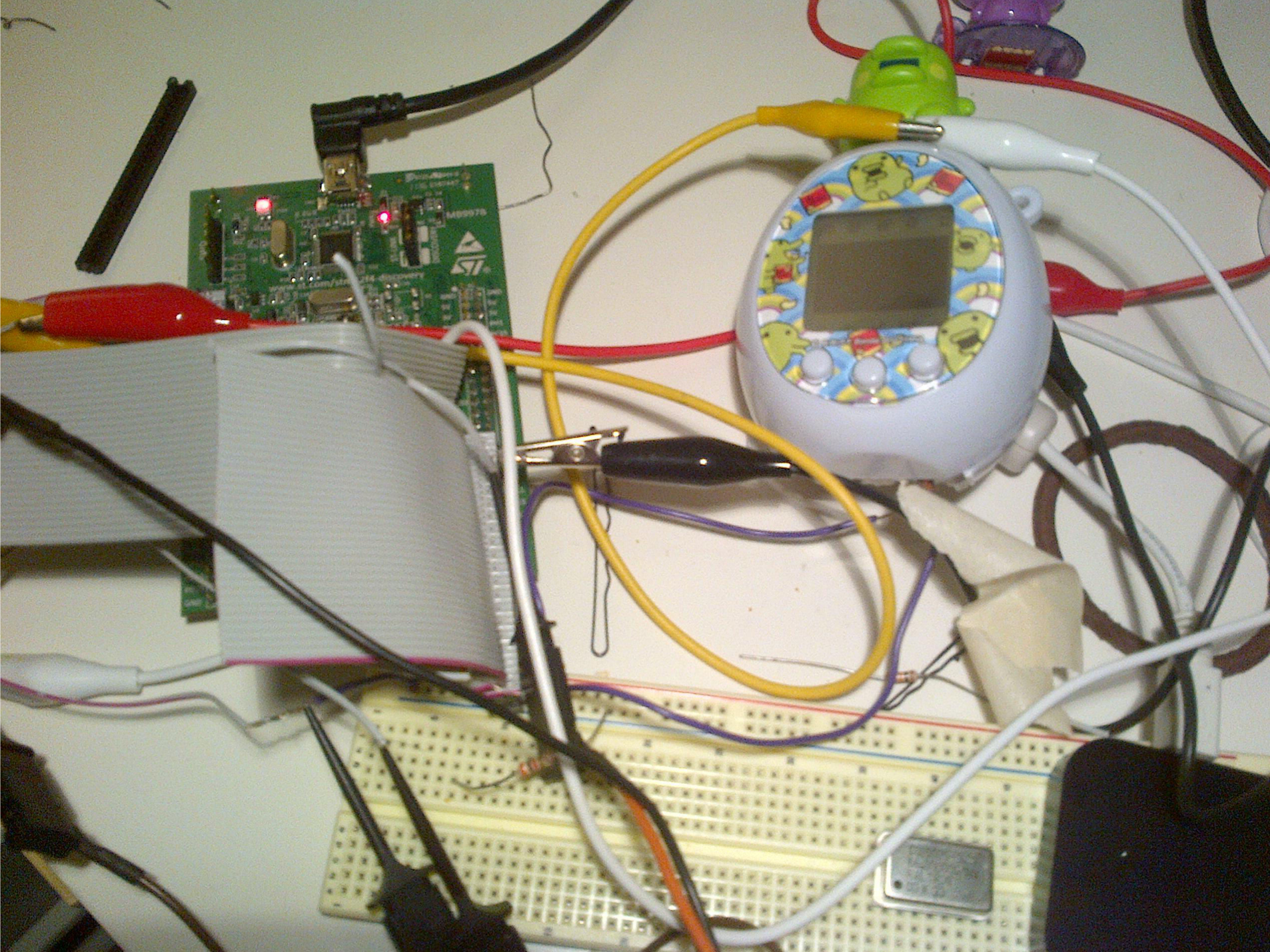
The Rest of the ROM

- The ROM contains a lot of non-image data
- None of this data is GeneralPlus code
  - Wrote a dissasembler
- Likely logic information in some sort of serialized format



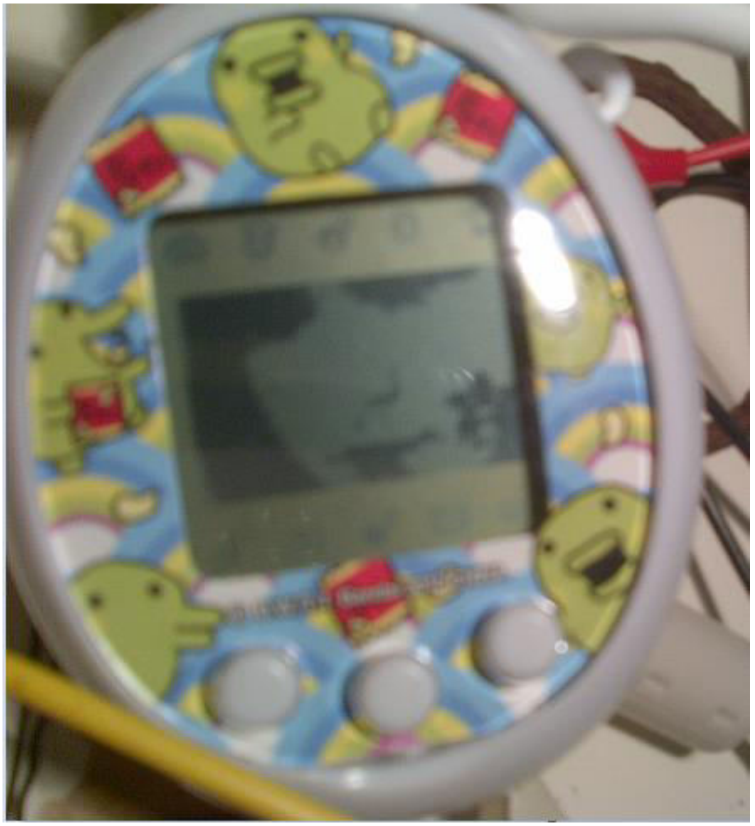
Simulating the ROM

- Could not obtain compatible flash
- Attempted to simulate the ROM using an Arduino, but chip is too slow
- Switched to a Chipkit Uno, this was also too slow
- Eventually used a STM32F4 Discovery board



Simulating the ROM

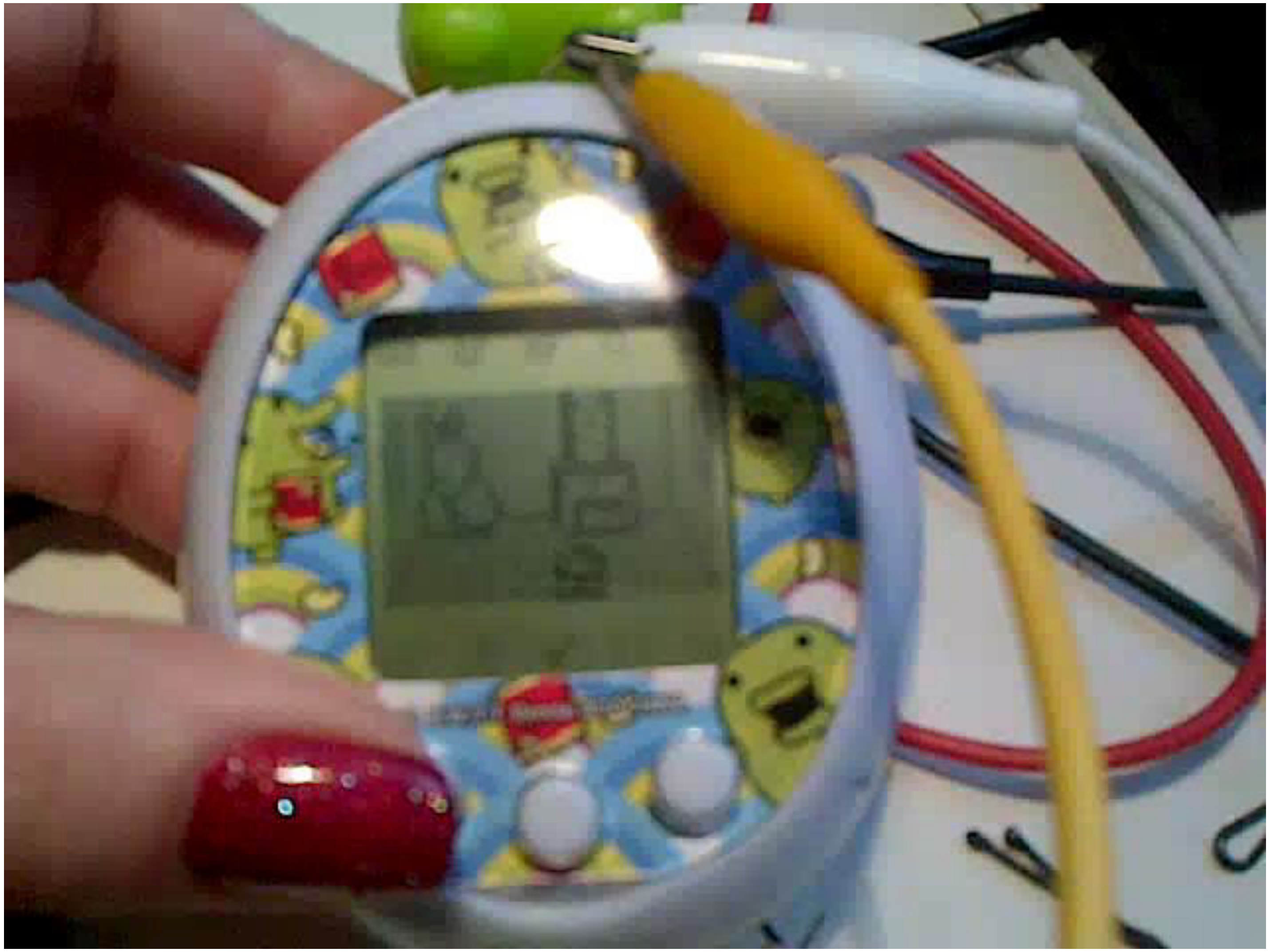
- Knew the image format, so could alter images



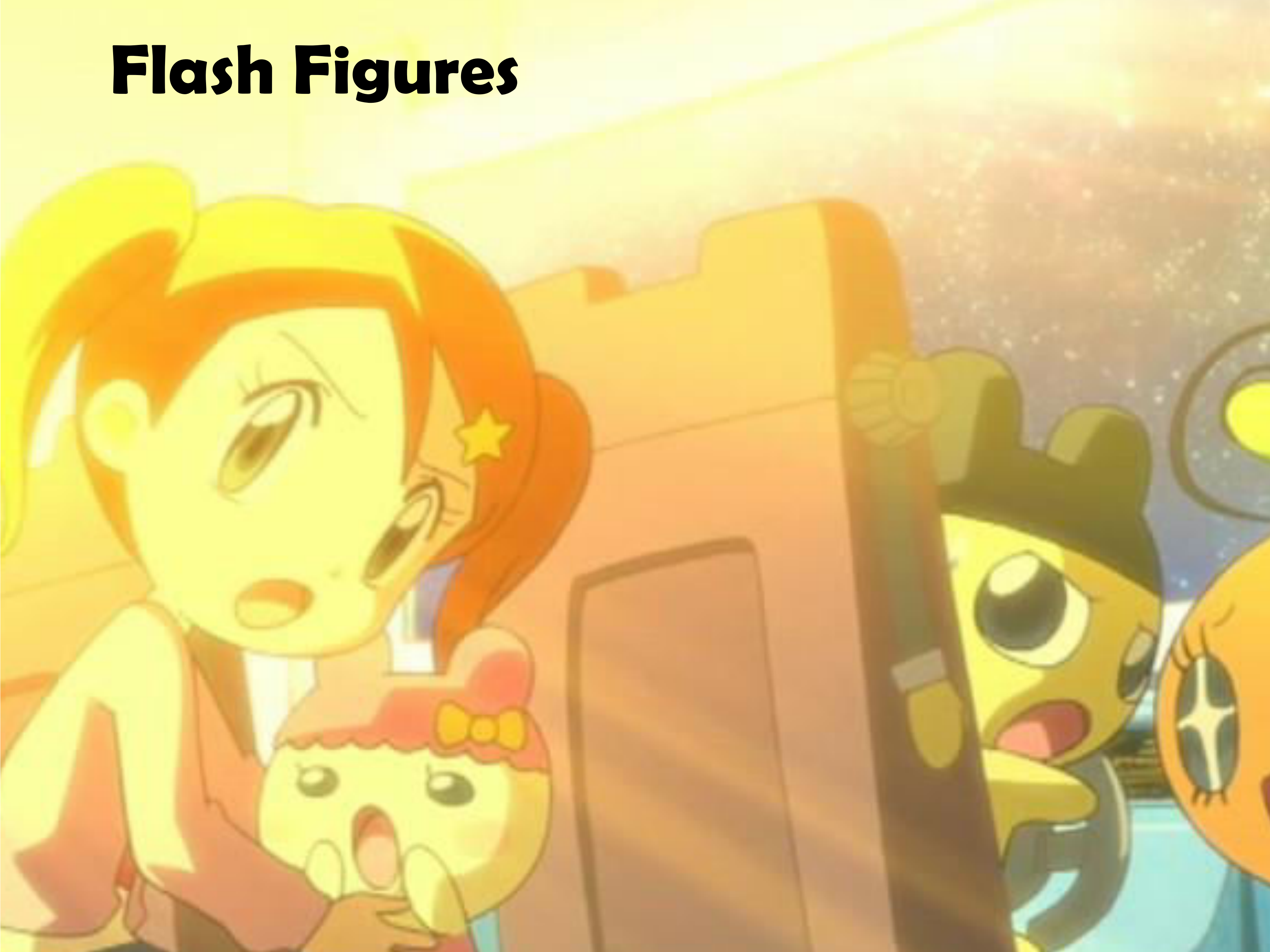
Game Logic

- The Tama-Go reads less than 50 bytes of non-image data during all figure functionality
- Game logic is represented by a one byte code
  - This logic is executed with images from figure
- Changing this code can cause a jump to non-game screens
  - Stats, food, death, etc. Every screen was available
- Many codes caused freezing

Evolve Demo

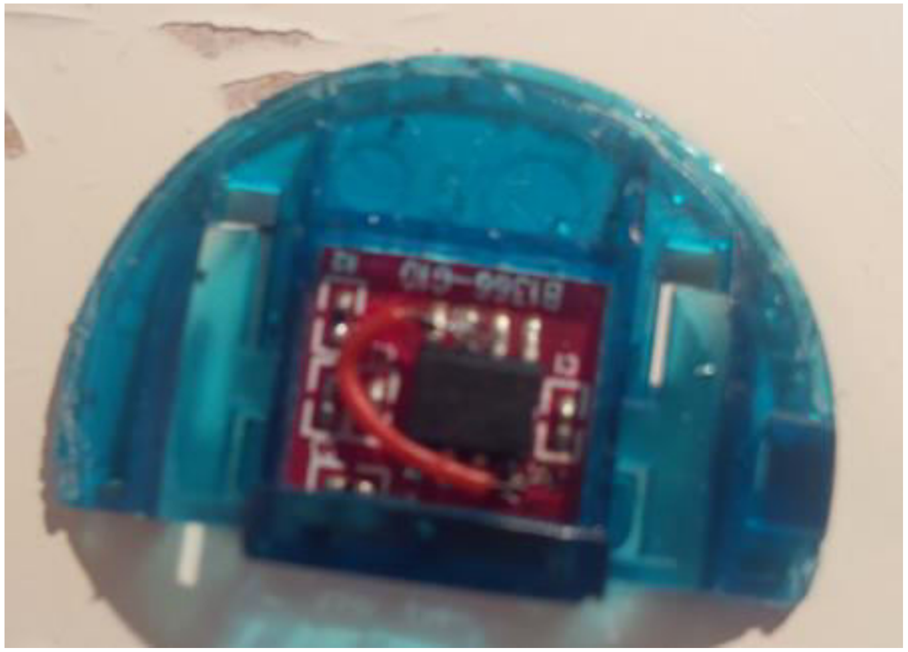


Flash Figures

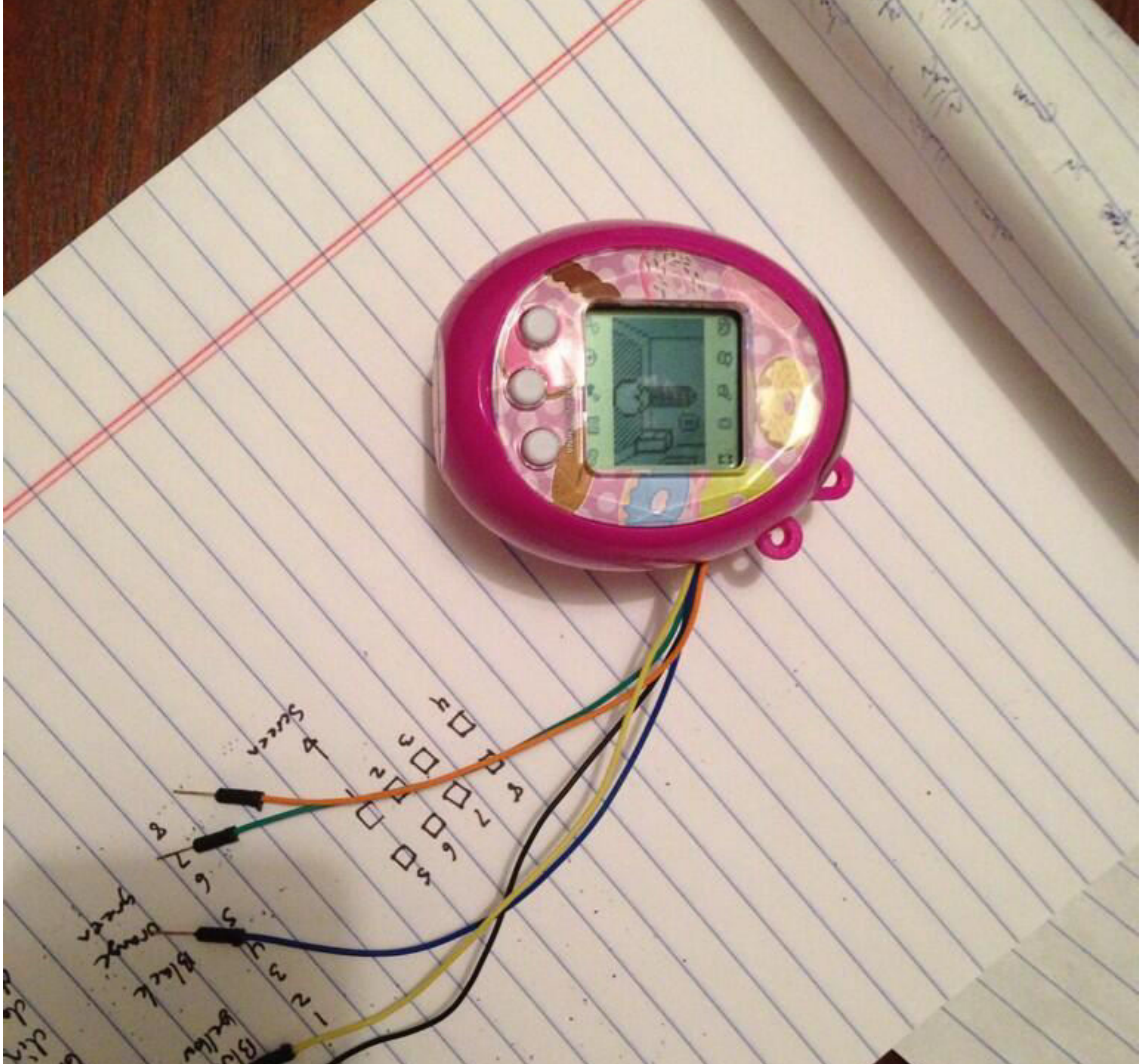


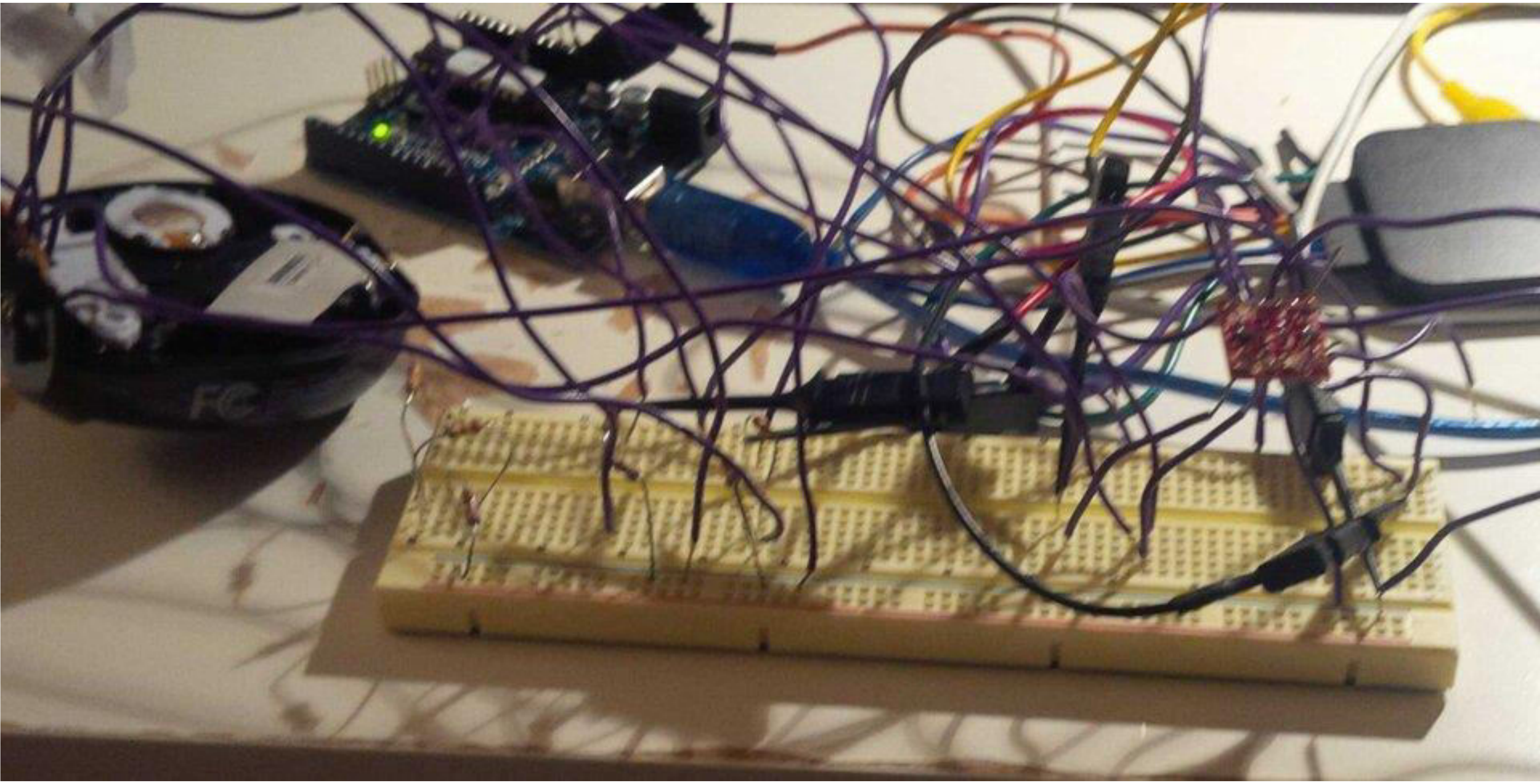
Flash Figures

- MrBlinky ordered a set of figures to experiment with
  - They contained flash!
  - Built a figure programmer
  - The ability to re-flash figures made testing much easier



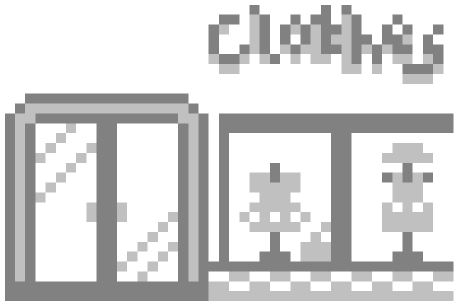
KRAFT
ALKALINE
MICRO • LR03 • AAA • 1.5V
ALKALINE
Ne pas recharger ni jeter au feu. Bien respecter le sens du branchement. Do not recharge

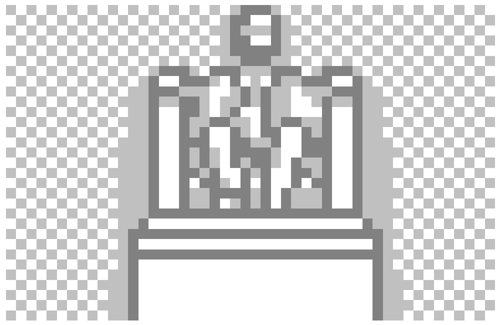

Screen
Black
orange
green
yellow
Blu

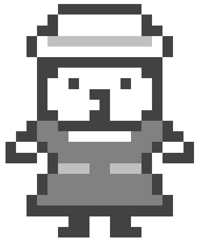


Items

- Items are implemented using a byte code format
  - Instructions include showing images, playing sounds and changing Tamagotchi stats
  - Some unusual behaviour for invalid instructions
  - Posted ‘dev tools’ on github







Demo

Code Execution

Game Logic

- The Tama-Go reads less than 50 bytes of non-image data during all figure functionality
- Game logic is represented by a one byte code
  - This logic is executed with images from figure
- Changing this code can cause a jump to non-game screens
  - Stats, food, death, etc. Every screen was available
- Many codes caused freezing

6502 Facts

- Memory mapped into a single address space
- No MMU
  - Unmapped addresses return 0 (usually)
  - Invalid instructions execute undefined behaviour
- Reset is rare
  - Great for explotation

First Attempt

- Assumed 'game codes' were indexes into a jump table
  - Invalid indexes would cause jumps (RTS) to non-pointer data
- Only controllable RAM is LCD RAM
  - 0x1000-0x1200
- Made a NOP sled and hoped

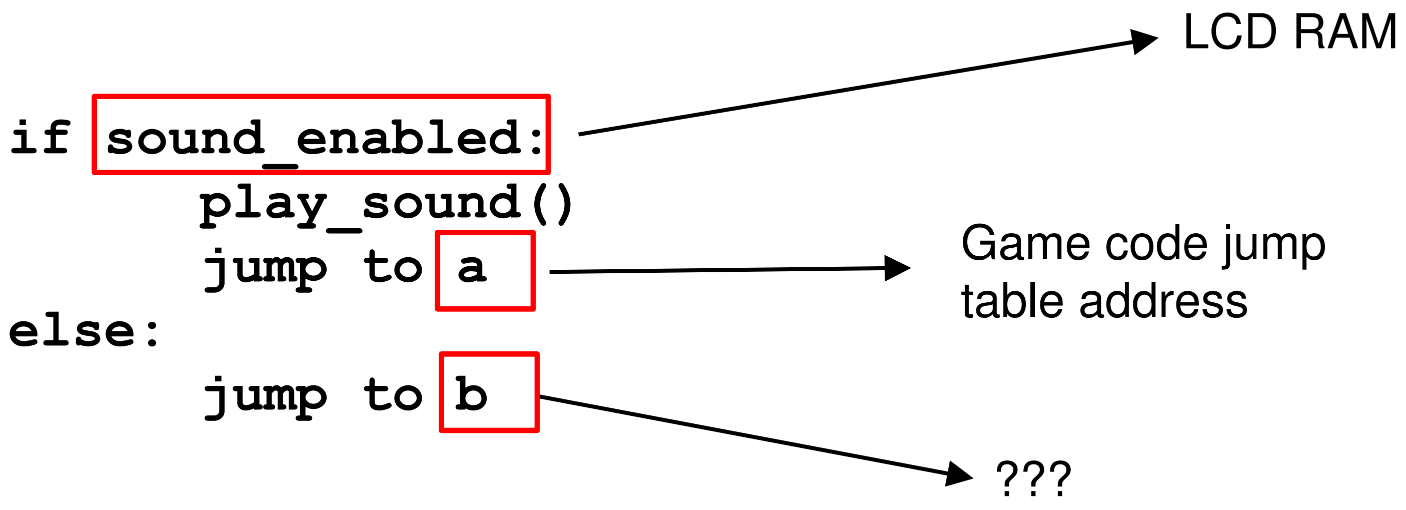
© B.W. / T.T. Bandai 2010 CHINA

Code 0xCC

- Did not work, but code 0xCC had interesting behavior
  - Buzzed when bit 3 of byte 68 was set and detected figure detach
  - Froze otherwise
- Also noticed that some middle indexes worked

New Theory

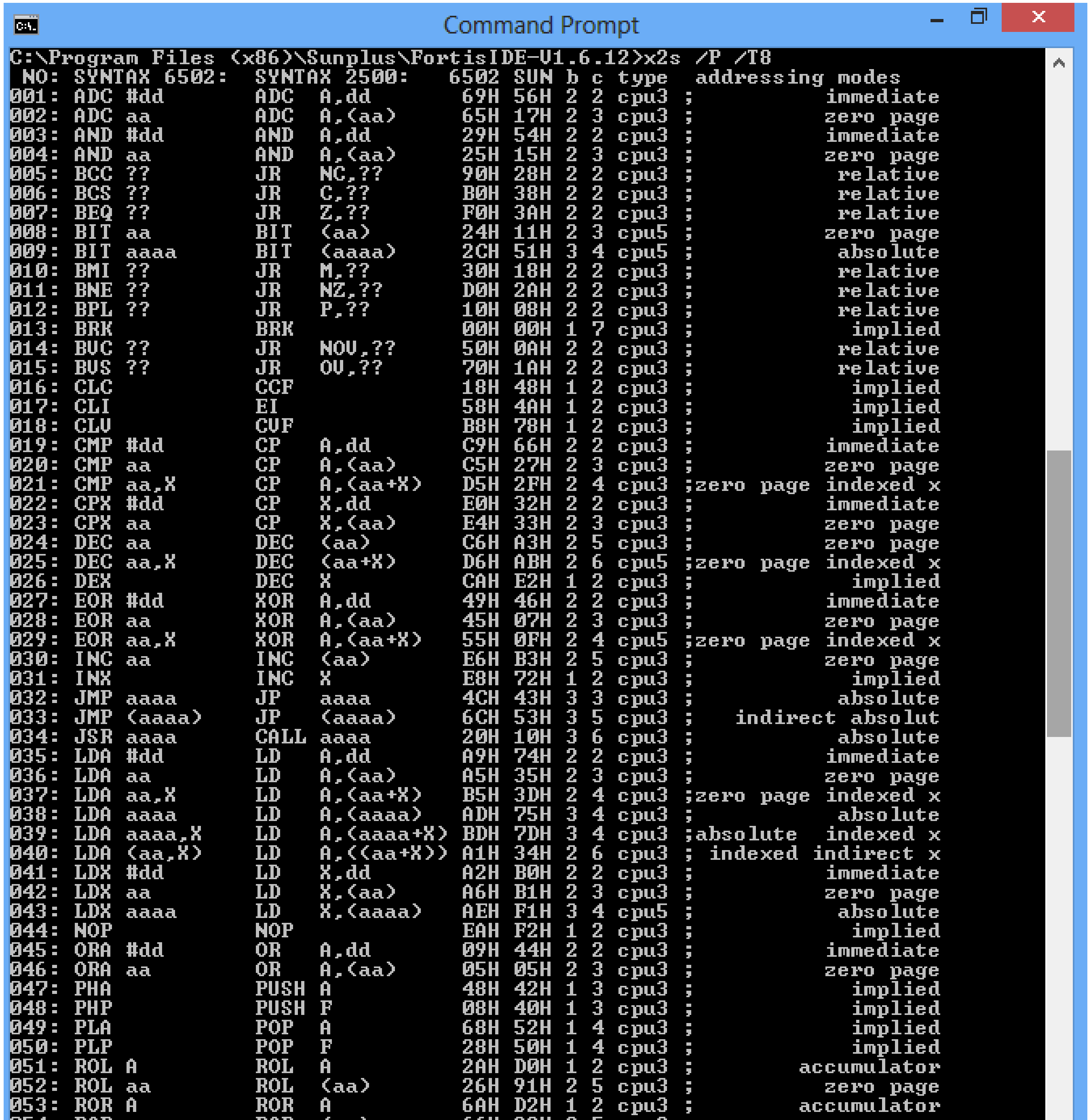
- All indexes are valid, but the stack isn't set up correctly
- 0xCC plays the noise when button pressed



```
if sound_enabled:
        play_sound()
        jump to a
else:
        jump to b
```

New Theory

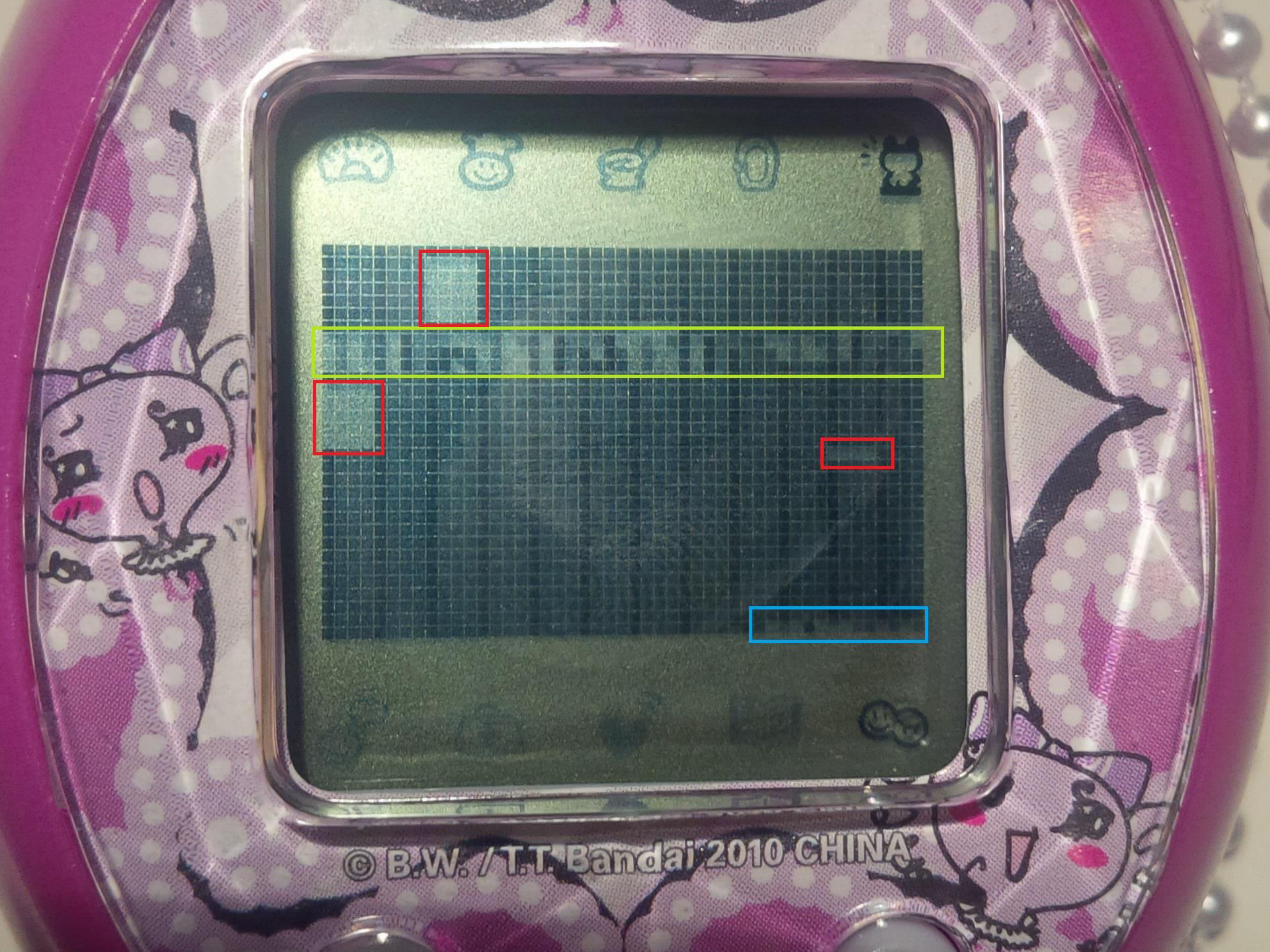
- But if
  - A pointer to the LCD RAM is on the stack
  - Stack confusion is occurring
  - There's 255 possibilities
- Why isn't it working?

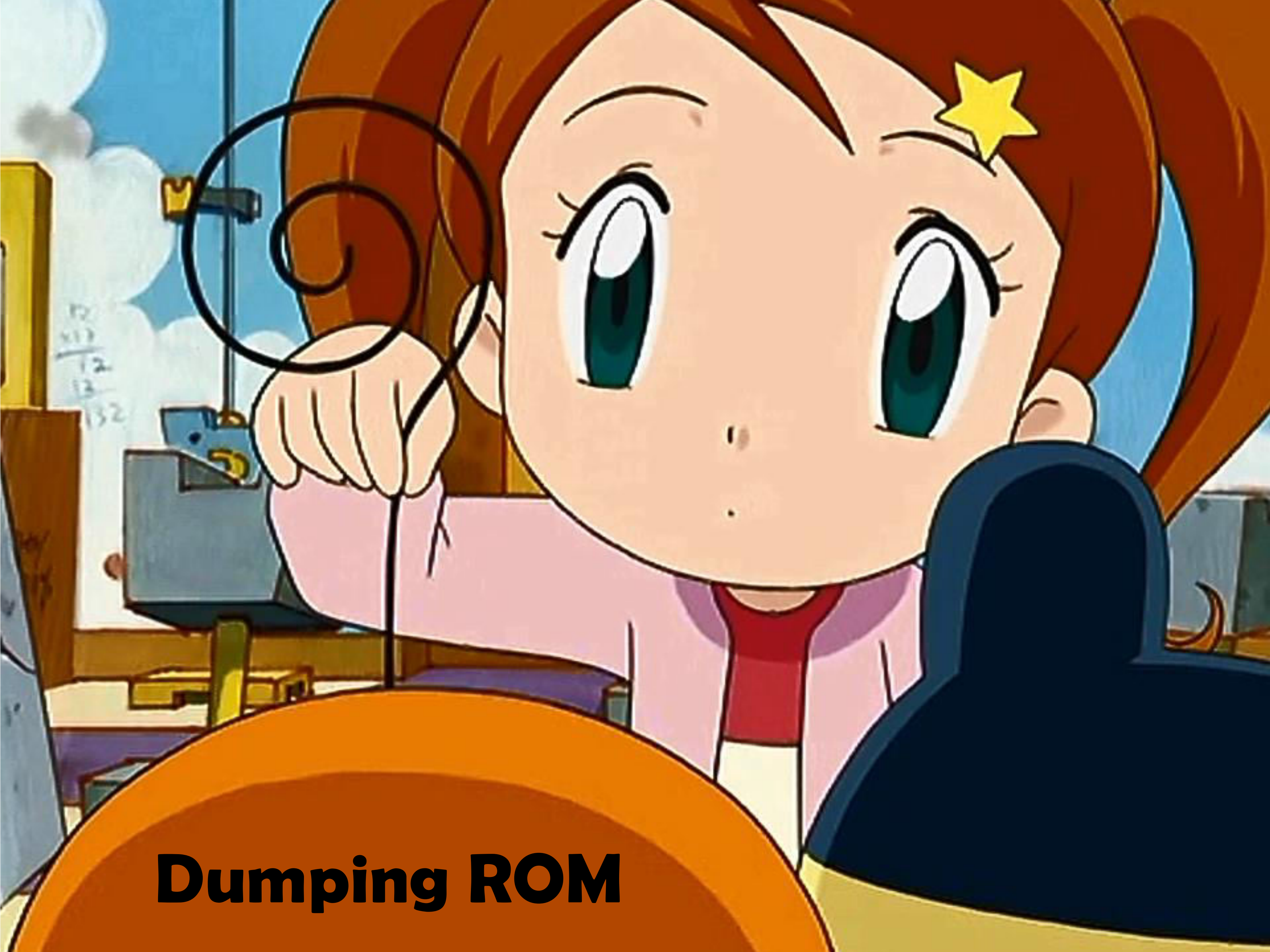


```
C:\Program Files (x86)\Sunplus\FortisIDE-V1.6.12>x2s /P /T8
 NO: SYNTAX 6502:    SYNTAX 2500:    6502 SUN b c type  addressing modes
001: ADC #dd         ADC  A,dd        69H 56H 2 2 cpu3 ;           immediate
002: ADC aa          ADC  A,(aa)      65H 17H 2 3 cpu3 ;           zero page
003: AND #dd         AND  A,dd        29H 54H 2 2 cpu3 ;           immediate
004: AND aa          AND  A,(aa)      25H 15H 2 3 cpu3 ;           zero page
005: BCC ??          JR   NC,??       90H 28H 2 2 cpu3 ;            relative
006: BCS ??          JR   C,??        B0H 38H 2 2 cpu3 ;            relative
007: BEQ ??          JR   Z,??        F0H 3AH 2 2 cpu3 ;            relative
008: BIT aa          BIT  (aa)        24H 11H 2 3 cpu5 ;           zero page
009: BIT aaaa        BIT  (aaaa)      2CH 51H 3 4 cpu5 ;            absolute
010: BMI ??          JR   M,??        30H 18H 2 2 cpu3 ;            relative
011: BNE ??          JR   NZ,??       D0H 2AH 2 2 cpu3 ;            relative
012: BPL ??          JR   P,??        10H 08H 2 2 cpu3 ;            relative
013: BRK             BRK              00H 00H 1 7 cpu3 ;             implied
014: BVC ??          JR   NOV,??      50H 0AH 2 2 cpu3 ;            relative
015: BVS ??          JR   OV,??       70H 1AH 2 2 cpu3 ;            relative
016: CLC             CCF              18H 48H 1 2 cpu3 ;             implied
017: CLI             EI               58H 4AH 1 2 cpu3 ;             implied
018: CLV             CVF              B8H 78H 1 2 cpu3 ;             implied
019: CMP #dd         CP   A,dd        C9H 66H 2 2 cpu3 ;           immediate
020: CMP aa          CP   A,(aa)      C5H 27H 2 3 cpu3 ;           zero page
021: CMP aa,X        CP   A,(aa+X)    D5H 2FH 2 4 cpu3 ;zero page  indexed x
022: CPX #dd         CP   X,dd        E0H 32H 2 2 cpu3 ;           immediate
023: CPX aa          CP   X,(aa)      E4H 33H 2 3 cpu3 ;           zero page
024: DEC aa          DEC  (aa)        C6H A3H 2 5 cpu3 ;           zero page
025: DEC aa,X        DEC  (aa+X)      D6H ABH 2 6 cpu5 ;zero page  indexed x
026: DEX             DEC  X           CAH E2H 1 2 cpu3 ;             implied
027: EOR #dd         XOR  A,dd        49H 46H 2 2 cpu3 ;           immediate
028: EOR aa          XOR  A,(aa)      45H 07H 2 3 cpu3 ;           zero page
029: EOR aa,X        XOR  A,(aa+X)    55H 0FH 2 4 cpu5 ;zero page  indexed x
030: INC aa          INC  (aa)        E6H B3H 2 5 cpu3 ;           zero page
031: INX             INC  X           E8H 72H 1 2 cpu3 ;             implied
032: JMP aaaa        JP   aaaa        4CH 43H 3 3 cpu3 ;            absolute
033: JMP (aaaa)      JP   (aaaa)      6CH 53H 3 5 cpu3 ;    indirect absolut
034: JSR aaaa        CALL aaaa        20H 10H 3 6 cpu3 ;            absolute
035: LDA #dd         LD   A,dd        A9H 74H 2 2 cpu3 ;           immediate
036: LDA aa          LD   A,(aa)      A5H 35H 2 3 cpu3 ;           zero page
037: LDA aa,X        LD   A,(aa+X)    B5H 3DH 2 4 cpu3 ;zero page  indexed x
038: LDA aaaa        LD   A,(aaaa)    ADH 75H 3 4 cpu3 ;            absolute
039: LDA aaaa,X      LD   A,(aaaa+X)  BDH 7DH 3 4 cpu3 ;absolute   indexed x
040: LDA (aa,X)      LD   A,((aa+X))  A1H 34H 2 6 cpu3 ; indexed indirect x
041: LDX #dd         LD   X,dd        A2H B0H 2 2 cpu3 ;           immediate
042: LDX aa          LD   X,(aa)      A6H B1H 2 3 cpu3 ;           zero page
043: LDX aaaa        LD   X,(aaaa)    AEH F1H 3 4 cpu5 ;            absolute
044: NOP             NOP              EAH F2H 1 2 cpu3 ;             implied
045: ORA #dd         OR   A,dd        09H 44H 2 2 cpu3 ;           immediate
046: ORA aa          OR   A,(aa)      05H 05H 2 3 cpu3 ;           zero page
047: PHA             PUSH A           48H 42H 1 3 cpu3 ;             implied
048: PHP             PUSH F           08H 40H 1 3 cpu3 ;             implied
049: PLA             POP  A           68H 52H 1 4 cpu3 ;             implied
050: PLP             POP  F           28H 50H 1 4 cpu3 ;             implied
051: ROL A           ROL  A           2AH D0H 1 2 cpu3 ;         accumulator
052: ROL aa          ROL  (aa)        26H 91H 2 5 cpu3 ;           zero page
053: ROR A           ROR  A           6AH D2H 1 2 cpu3 ;         accumulator
```

Code Execution

- Switched instruction sets
- Used simpler shellcode
- Using the correct instruction set, it worked on the fourth index I tried, 0xd4


© B.W. / T.T. Bandai 2010 CHINA



Dumping ROM

Dumping Memory

- Wrote code to dump entire memory space of Tamagotchi
- Output memory over SPI using port A (buttons)
- Decoded output with signal analyzer

Paging

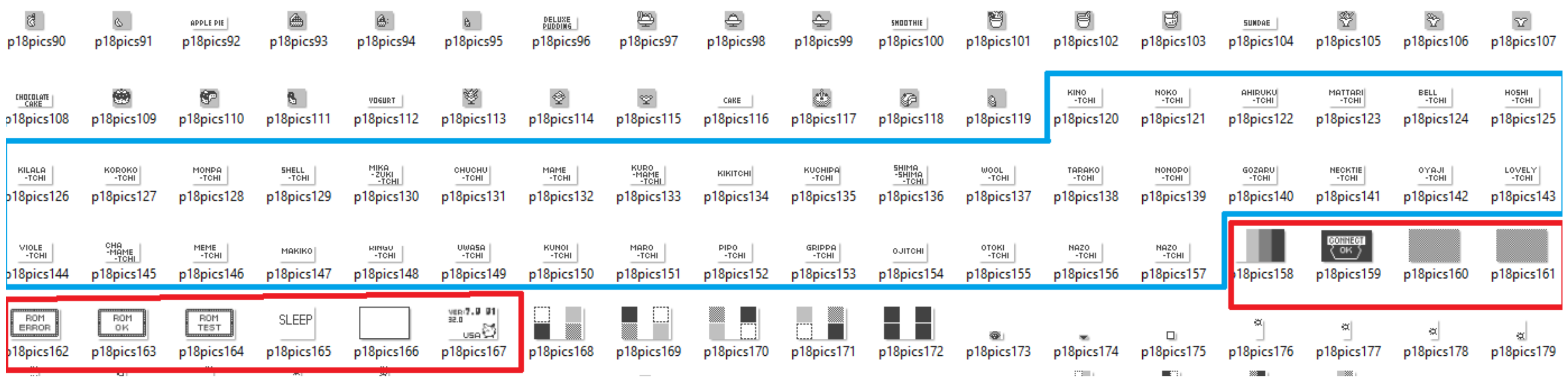
- The ROM is larger than the memory space
- First page is always mapped
- Other pages are mapped one at a time
- Determined 0x3000 is page port
- Dumped all 19 pages

Pages

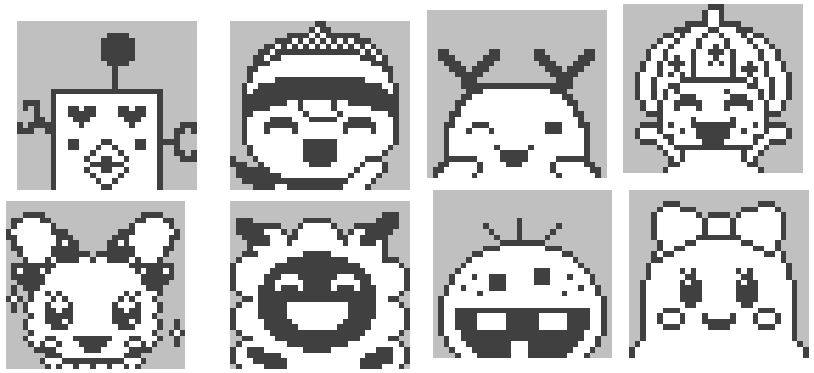
- Quickly identified pages by inspection
  - Pages 0 to 6 are code
  - Pages 7 to 9 are blank
  - Page 10 contains images and a image pointer table
  - Pages 11 to 18 contain image data
  - Page 19 contains audio

Images

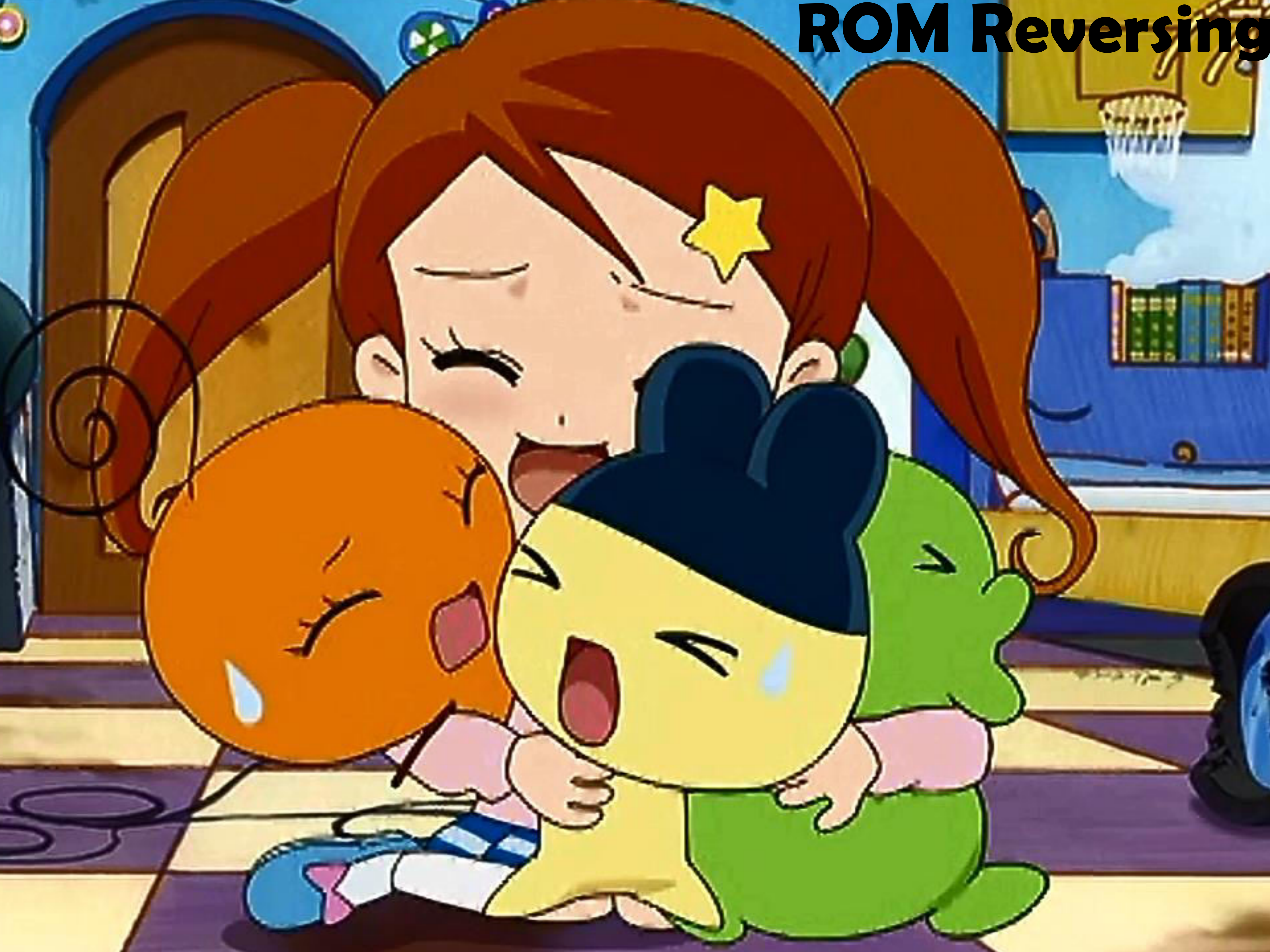
- Dumped images from image pages



```
p18pics90   p18pics91   p18pics92   p18pics93   p18pics94   p18pics95   p18pics96   p18pics97   p18pics98   p18pics99   p18pics100  p18pics101  p18pics102  p18pics103  p18pics104  p18pics105  p18pics106  p18pics107
p18pics108  p18pics109  p18pics110  p18pics111  p18pics112  p18pics113  p18pics114  p18pics115  p18pics116  p18pics117  p18pics118  p18pics119  p18pics120  p18pics121  p18pics122  p18pics123  p18pics124  p18pics125
p18pics126  p18pics127  p18pics128  p18pics129  p18pics130  p18pics131  p18pics132  p18pics133  p18pics134  p18pics135  p18pics136  p18pics137  p18pics138  p18pics139  p18pics140  p18pics141  p18pics142  p18pics143
p18pics144  p18pics145  p18pics146  p18pics147  p18pics148  p18pics149  p18pics150  p18pics151  p18pics152  p18pics153  p18pics154  p18pics155  p18pics156  p18pics157  p18pics158  p18pics159  p18pics160  p18pics161
p18pics162  p18pics163  p18pics164  p18pics165  p18pics166  p18pics167  p18pics168  p18pics169  p18pics170  p18pics171  p18pics172  p18pics173  p18pics174  p18pics175  p18pics176  p18pics177  p18pics178  p18pics179
```

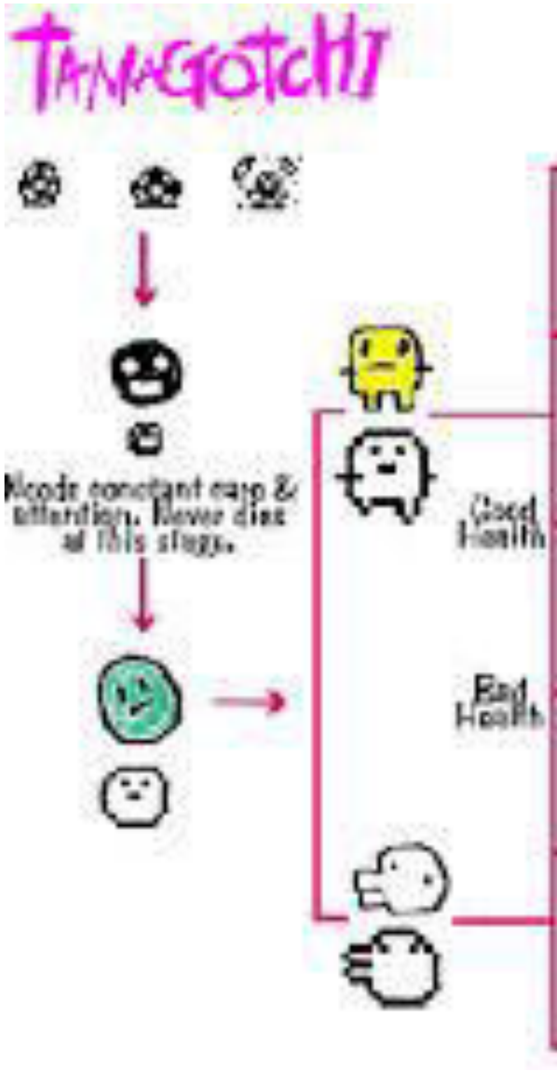


ROM Reversing



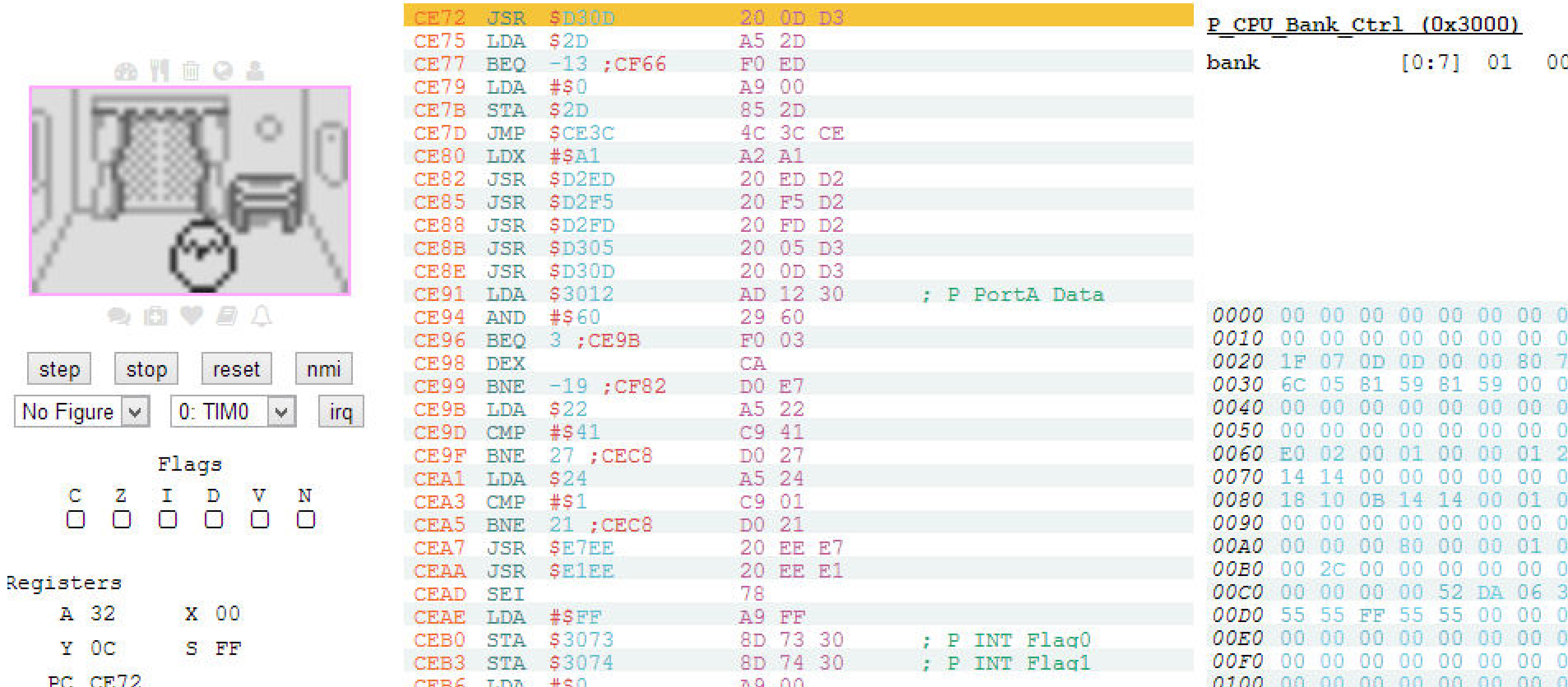
ROM Reversing

- Started using IDA
  - Learning curve was steep
  - No paging support
- Eventually wrote a simulator based on py65
  - Added support for LCD and ports
  - Slowly decoded the secrets of Tamagotchi life



Better Emulator

- Asterick wrote a JavaScript-based emulator
  - https://github.com/asterick/tamago



```
Flags
C  Z  I  D  V  N

Registers
A 32     X 00
Y 0C     S FF
PC CE72
```

```
CE72  JSR  $D30D        20 0D D3
CE75  LDA  $2D          A5 2D
CE77  BEQ  -13 ;CF66    F0 ED
CE79  LDA  #$0          A9 00
CE7B  STA  $2D          85 2D
CE7D  JMP  $CE3C        4C 3C CE
CE80  LDX  #$A1         A2 A1
CE82  JSR  $D2ED        20 ED D2
CE85  JSR  $D2F5        20 F5 D2
CE88  JSR  $D2FD        20 FD D2
CE8B  JSR  $D305        20 05 D3
CE8E  JSR  $D30D        20 0D D3
CE91  LDA  $3012        AD 12 30      ; P PortA Data
CE94  AND  #$60         29 60
CE96  BEQ  3 ;CE9B      F0 03
CE98  DEX               CA
CE99  BNE  -19 ;CF82    D0 E7
CE9B  LDA  $22          A5 22
CE9D  CMP  #$41         C9 41
CE9F  BNE  27 ;CEC8     D0 27
CEA1  LDA  $24          A5 24
CEA3  CMP  #$1          C9 01
CEA5  BNE  21 ;CEC8     D0 21
CEA7  JSR  $E7EE        20 EE E7
CEAA  JSR  $E1EE        20 EE E1
CEAD  SEI               78
CEAE  LDA  #$FF         A9 FF
CEB0  STA  $3073        8D 73 30      ; P INT Flag0
CEB3  STA  $3074        8D 74 30      ; P INT Flag1
CEB6  LDA  #$0          A9 00
```

```
P_CPU_Bank_Ctrl (0x3000)
bank              [0:7]  01   00

0000 00 00 00 00 00 00 00 0
0010 00 00 00 00 00 00 00 0
0020 1F 07 0D 0D 00 00 80 7
0030 6C 05 81 59 81 59 00 0
0040 00 00 00 00 00 00 00 0
0050 00 00 00 00 00 00 00 0
0060 E0 02 00 01 00 00 01 2
0070 14 14 00 00 00 00 00 0
0080 18 10 0B 14 14 00 01 0
0090 00 00 00 00 00 00 00 0
00A0 00 00 00 80 00 00 01 0
00B0 00 2C 00 00 00 00 00 0
00C0 00 00 00 00 52 DA 06 3
00D0 55 55 FF 55 55 00 00 0
00E0 00 00 00 00 00 00 00 0
00F0 00 00 00 00 00 00 00 0
0100 00 00 00 00 00 00 00 0
```

Tamagotchi Internals

- After start-up, Tamagotchis cycle through a single loop, driven by tm1 interrupts
- Always in one of 0x41 states
  - Table determines state actions
  - Can have substates and subsubstates and …
  - State entry behaves differently
  - States are responsible for all behaviour (buttons, sound) except for physical LCD update and SPI poll
  - A LOT of pointer tables

Secrets So Far …

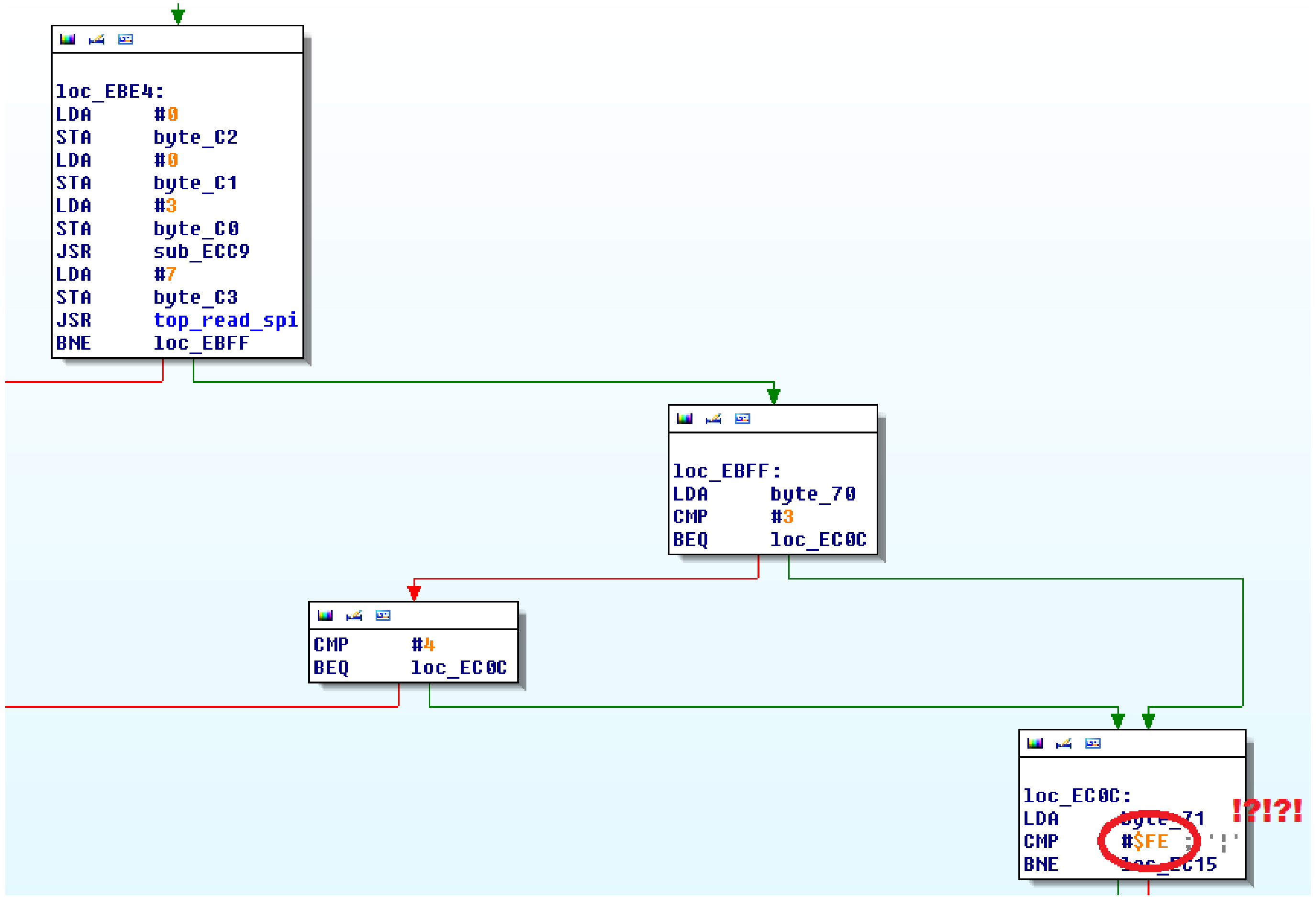
- What makes a Tamagotchi a boy or a girl?
  - Determined from entropy source C4, based on how many times tm1 has fired since the Tamagotchi started up
- What toddler a baby grows into is random
  - Intentionally evened out
  - Some toddlers are higher-maintenance than others

Secrets So Far …

- What teen a toddler becomes is based on care
  - Two factors
- What adult a teen becomes depends on care and training
  - Toddler care matters
- You can potty train your Tamagotchi

Test mode

- Uncovered a test mode if figure ID is 0xFE



```
loc_EBE4:
LDA      #0
STA      byte_C2
LDA      #0
STA      byte_C1
LDA      #3
STA      byte_C0
JSR      sub_ECC9
LDA      #7
STA      byte_C3
JSR      top_read_spi
BNE      loc_EBFF
```

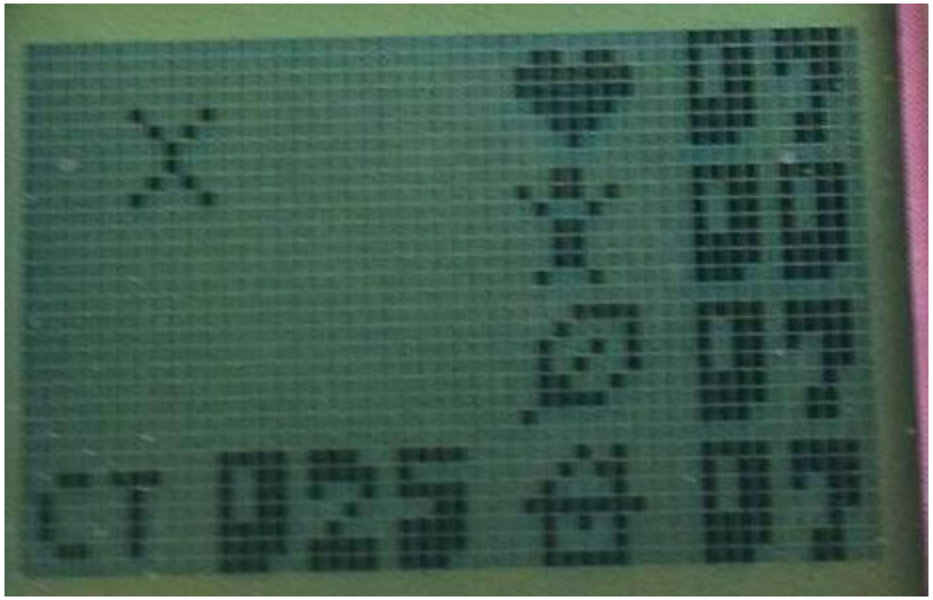
```
loc_EBFF:
LDA      byte_70
CMP      #3
BEQ      loc_EC0C
```

```
CMP      #4
BEQ      loc_EC0C
```

```
loc_EC0C:
LDA      byte_71
CMP      #$FE ; '¦'
BNE      loc_EC15
```

Test Mode

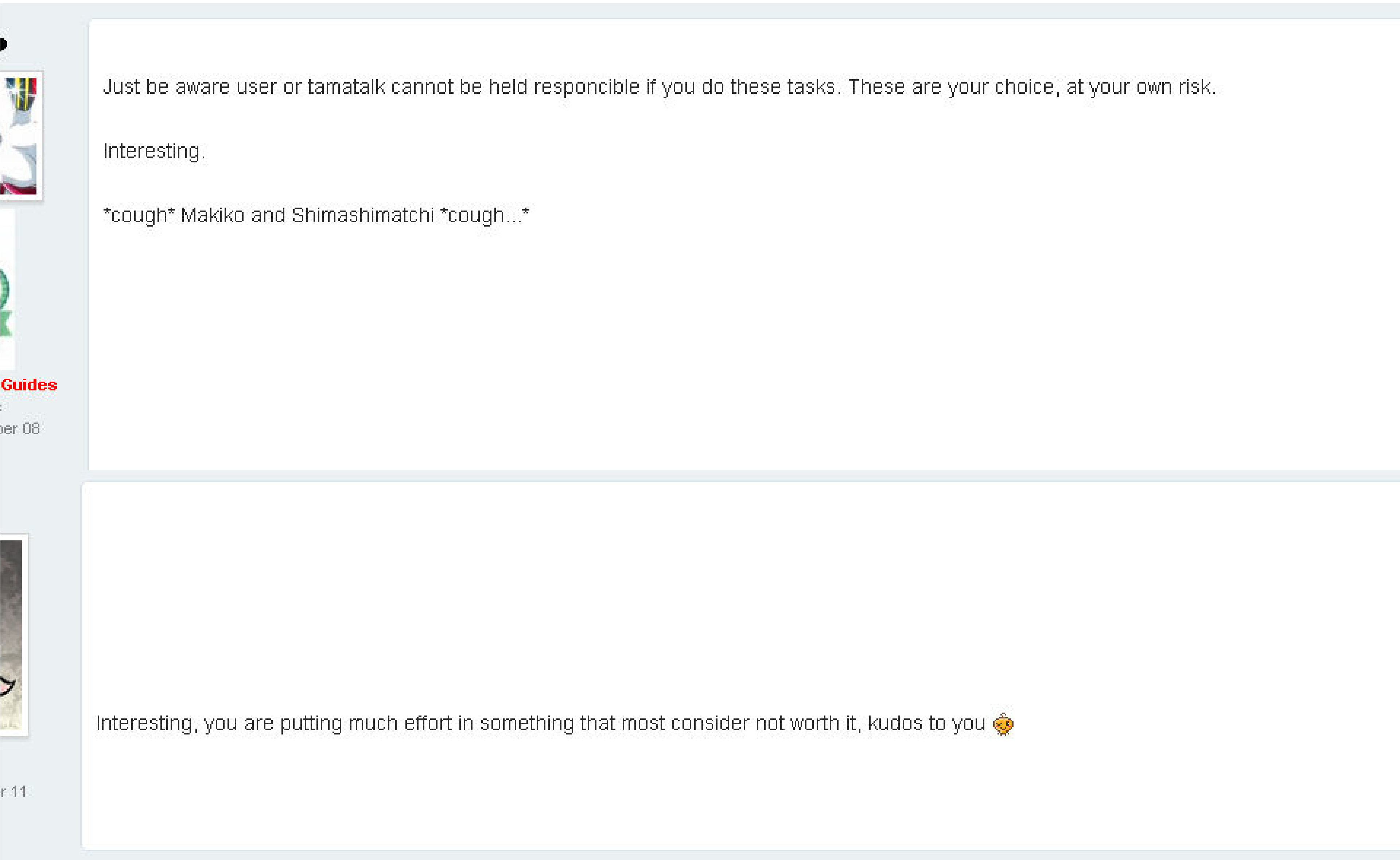
- Allows all stats to be altered
- Allows character and spouse to be selected
- Allows care factors to be viewed and altered
- Two unused care factors



More Secrets

- It doesn’t matter who your Tamagotchi marries
  - They’re just as happy
  - The kids turn out just the same
    - Unless you marry an Olditchi
- Figures don’t alter Tamagotchi functionality outside of their functionality
  - Special display for 100 figures

Reaction



Just be aware user or tamatalk cannot be held responcible if you do these tasks. These are your choice, at your own risk.

Interesting.

cough Makiko and Shimashimatchi *cough...*

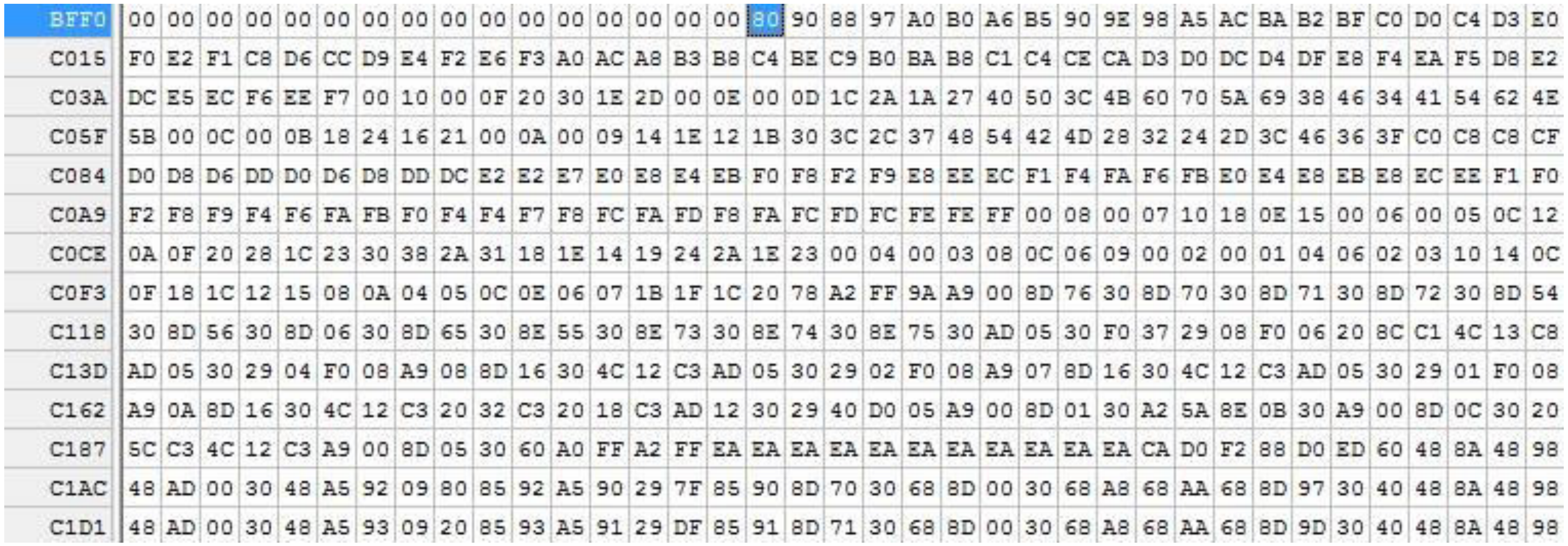
Interesting, you are putting much effort in something that most consider not worth it, kudos to you

HATERS
GONNA HATE

Test Program

GeneralPlus Test Program

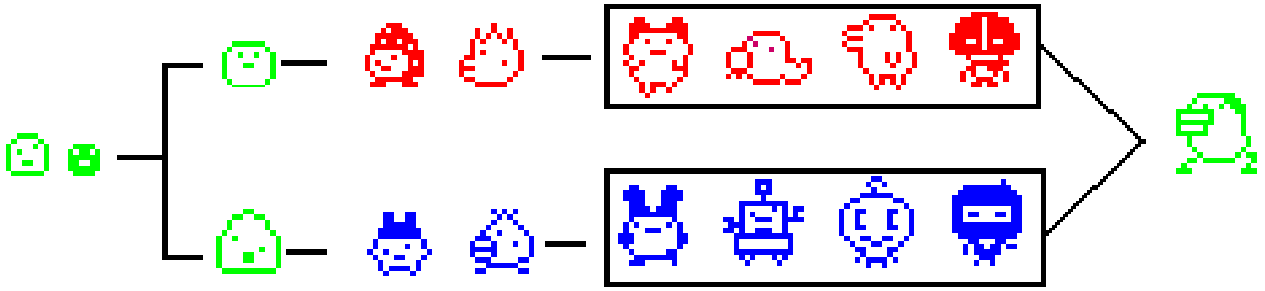
- Analyzed GeneralPlus Test Program
- Hoped it would make dumping other GP ROMs easier



```
BFF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 90 88 97 A0 B0 A6 B5 90 9E 98 A5 AC BA B2 BF C0 D0 C4 D3 E0
C015 F0 E2 F1 C8 D6 CC D9 E4 F2 E6 F3 A0 AC A8 B3 B8 C4 BE C9 B0 BA B8 C1 C4 CE CA D3 D0 DC D4 DF E8 F4 EA F5 D8 E2
C03A DC E5 EC F6 EE F7 00 10 00 0F 20 30 1E 2D 00 0E 00 0D 1C 2A 1A 27 40 50 3C 4B 60 70 5A 69 38 46 34 41 54 62 4E
C05F 5B 00 0C 00 0B 18 24 16 21 00 0A 00 09 14 1E 12 1B 30 3C 2C 37 48 54 42 4D 28 32 24 2D 3C 46 36 3F C0 C8 C8 CF
C084 D0 D8 D6 DD D0 D6 D8 DD DC E2 E2 E7 E0 E8 E4 EB F0 F8 F2 F9 E8 EE EC F1 F4 FA F6 FB E0 E4 E8 EB E8 EC EE F1 F0
C0A9 F2 F8 F9 F4 F6 FA FB F0 F4 F4 F7 F8 FC FA FD F8 FA FC FD FC FE FE FF 00 08 00 07 10 18 0E 15 00 06 00 05 0C 12
C0CE 0A 0F 20 28 1C 23 30 38 2A 31 18 1E 14 19 24 2A 1E 23 00 04 00 03 08 0C 06 09 00 02 00 01 04 06 02 03 10 14 0C
C0F3 0F 18 1C 12 15 08 0A 04 05 0C 0E 06 07 1B 1F 1C 20 78 A2 FF 9A A9 00 8D 76 30 8D 70 30 8D 71 30 8D 72 30 8D 54
C118 30 8D 56 30 8D 06 30 8D 65 30 8E 55 30 8E 73 30 8E 74 30 8E 75 30 AD 05 30 F0 37 29 08 F0 06 20 8C C1 4C 13 C8
C13D AD 05 30 29 04 F0 08 A9 08 8D 16 30 4C 12 C3 AD 05 30 29 02 F0 08 A9 07 8D 16 30 4C 12 C3 AD 05 30 29 01 F0 08
C162 A9 0A 8D 16 30 4C 12 C3 20 32 C3 20 18 C3 AD 12 30 29 40 D0 05 A9 00 8D 01 30 A2 5A 8E 0B 30 A9 00 8D 0C 30 20
C187 5C C3 4C 12 C3 A9 00 8D 05 30 60 A0 FF A2 FF EA EA EA EA EA EA EA EA EA EA EA CA D0 F2 88 D0 ED 60 48 8A 48 98
C1AC 48 AD 00 30 48 A5 92 09 80 85 92 A5 90 29 7F 85 90 8D 70 30 68 8D 00 30 68 A8 68 AA 68 8D 97 30 40 48 8A 48 98
C1D1 48 AD 00 30 48 A5 93 09 20 85 93 A5 91 29 DF 85 91 8D 71 30 68 8D 00 30 68 A8 68 AA 68 8D 9D 30 40 48 8A 48 98
```

GeneralPlus Test Program

- Polls port A for a code, runs test and outputs results on port B
- Two interesting codes, 3 and 0x16
- Code 3 checksums custom address range
  - Unfortunately contains a bug so it doesn't work



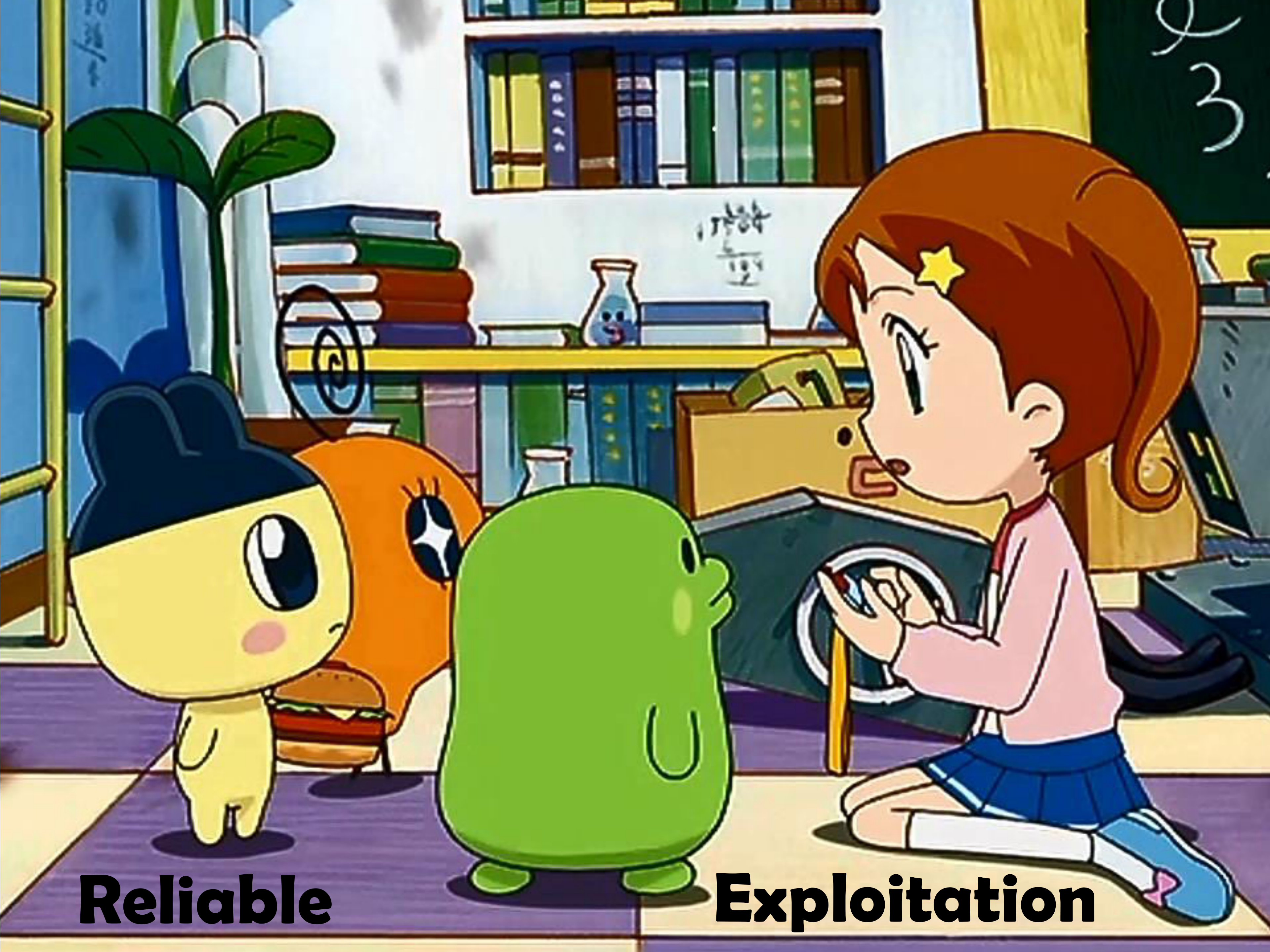
Test Program Code Dump

- Code 16 fills RAM up with code from Port B and jumps to it!
- Can dump code from any GeneralPlus LCD controller so long as Port A, Port B and TEST are bonded

Dev Tools

Existing Tools

- Wrote two ‘dev’ tools in the process of reversing
  - portrait.py puts an image on the Tamagotchi screen
  - itemmake.py makes a ‘music video’ based on a script
- Both have serious limitations
- Wanted to write a tools that allows generic 6502 execution

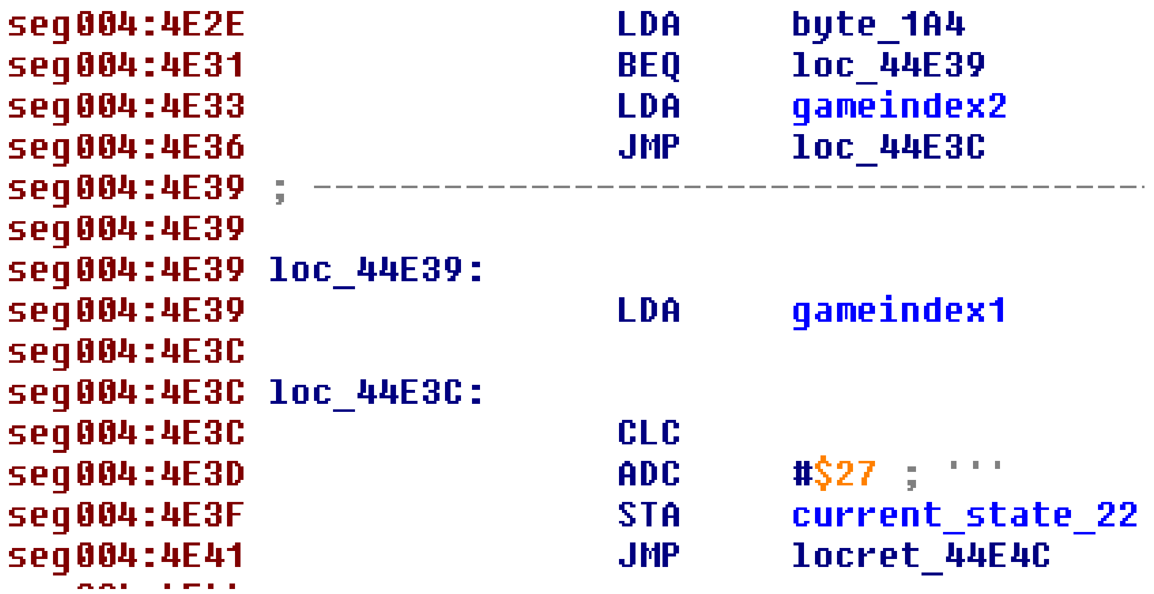

Reliable
Exploitation

Reliable Exploitation

- The vulnerability used to dump the ROM was 30-40% reliable
  - Worked better if the Tamagotchi had been running awhile
- Needed 100% reliability for a useful dev tool

The ROM Dump Vuln (D4)

- The game indices in the figure ROM cause a state change to 0x27 + the index



```
seg004:4E2E                 LDA     byte_1A4
seg004:4E31                 BEQ     loc_44E39
seg004:4E33                 LDA     gameindex2
seg004:4E36                 JMP     loc_44E3C
seg004:4E39 ; ---------------------------------------
seg004:4E39
seg004:4E39 loc_44E39:
seg004:4E39                 LDA     gameindex1
seg004:4E3C
seg004:4E3C loc_44E3C:
seg004:4E3C                 CLC
seg004:4E3D                 ADC     #$27 ; '''
seg004:4E3F                 STA     current_state_22
seg004:4E41                 JMP     locret_44E4C
```

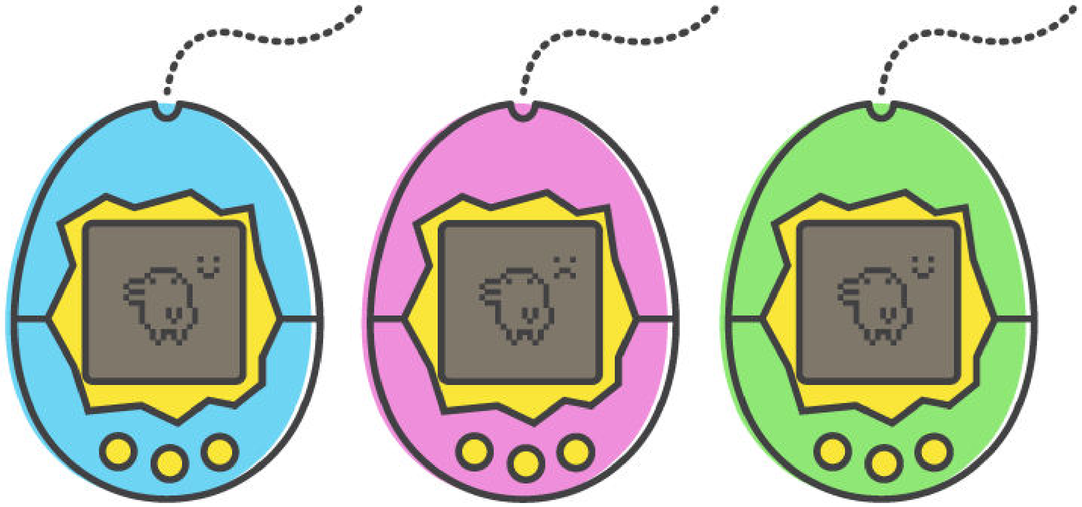
- Valid indices are between 0 and 0x41
  - No validity check

The ROM Dump Vuln (D4)

- On a state change
  - Tamagotchi indexes into a state page table, switches to the page at the index and jumps to 0x4000
  - Code pages have code at 0x4000 that indexes into a jump table for the page
  - Invalid states could cause a jump to a non-code page, or a jump to an unexpected address

The ROM Dump Vuln (D4)

- State is set to 0x27 + 0xD4 (0xFB)
  - Page table returns 0x3c (actually part of LCD table)
- Switching to page 0x3c makes memory at 0x4000 float
  - No wonder this exploit is unreliable



Vulnerability Idol

- Finding a more reliable index required a lot of tracing
- Eventually tried several indexes to find one that seemed reliable
  - 0xCD was a good contender

Index 0xCD

- State is set to 0x27 + 0xCD (0xF4)
  - Page table returns 0x4 (also part of LCD table)
- Loads page 4 and indexes jump table at 0xF4
  - This location is actually code: INC $11E
  - As data, it resolves to location 0x1EEE
  - LCD RAM addressing ignores bits 2-7 of byte 3
  - Resolves to 0x10EE (in LCD RAM)
- This exploit will always work

Dev Kit

tASMgotchi

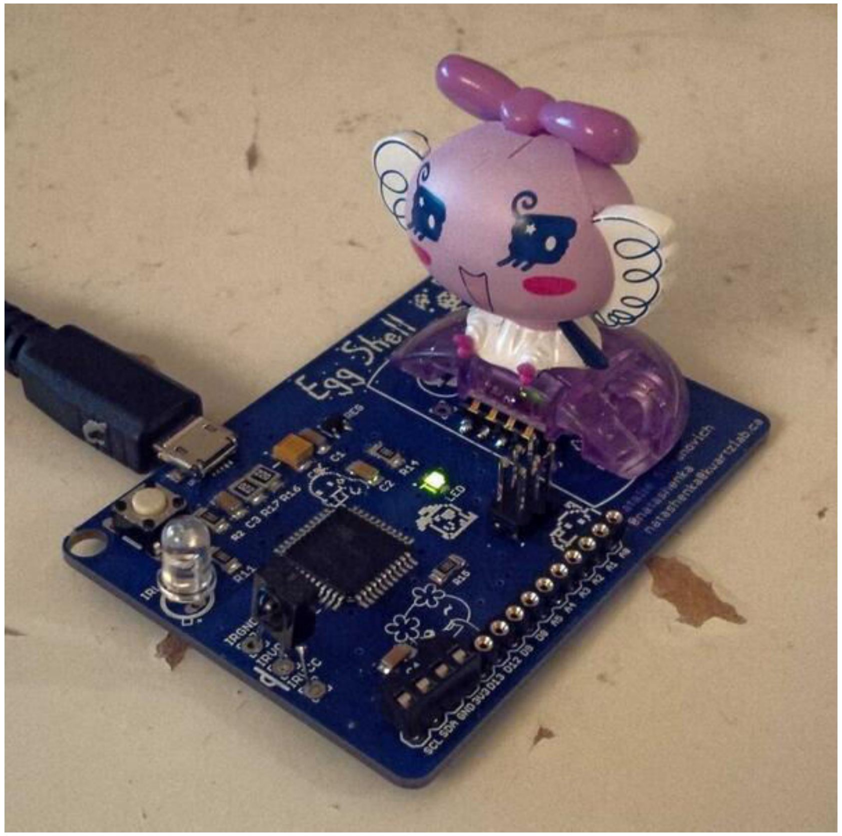
- 6502 Assembler for Tamagotchi
- Outputs binary ready to be loaded on figure
- Loads code into RAM, and automatically handles paging during execution
- Contains convenience functions for common functionality such as LCD writes and IR
  - Largely from Tamagotchi ROM
- Ophis based

Making the Dev Kit

- Lack of datasheet made writing some functions difficult
  - Limited knowledge of port locations
- Determined a lot of functionality from the test program
- Still a lot of unknowns
  - Power management, SPU, watchdog
  - Contributions welcome!

Making the Dev Kit

- Egg Shell board
- SPI programmer and IR for future RCE ☺
- Also a Lilypad USB Arduino



Tamagotchi Tools

https://github.com/natashenka/Egg-Shell

- Portrait maker
- Item maker
- tASMgotchi
- Board specs

Workshop

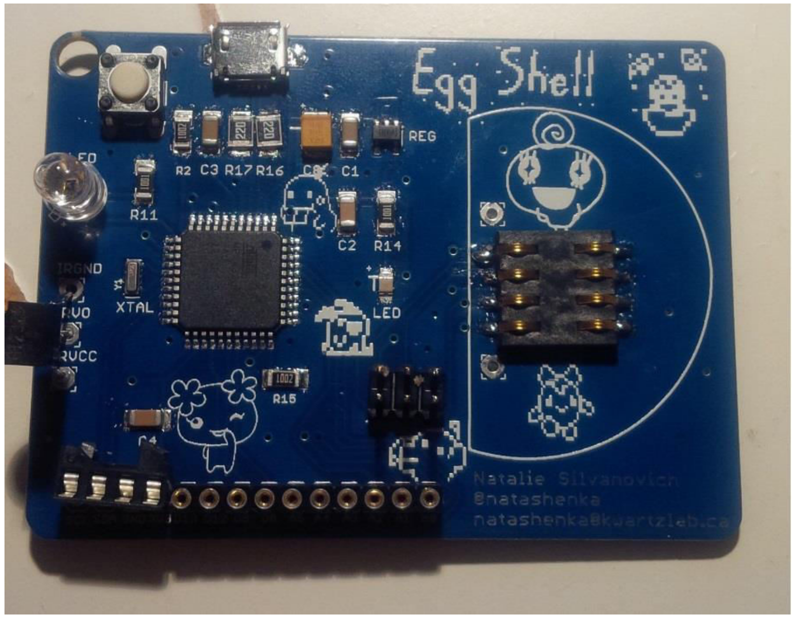
Learn to hack Tamagotchis here at 30c3!

Today at 7:30pm in Hall E

Kit is €25 + VAT, and includes a Tamagotchi, figure and a programming board

Egg Shell Boards

- Boards €11, PCBs €2



- http://natashenka.ca/boards/

Demo

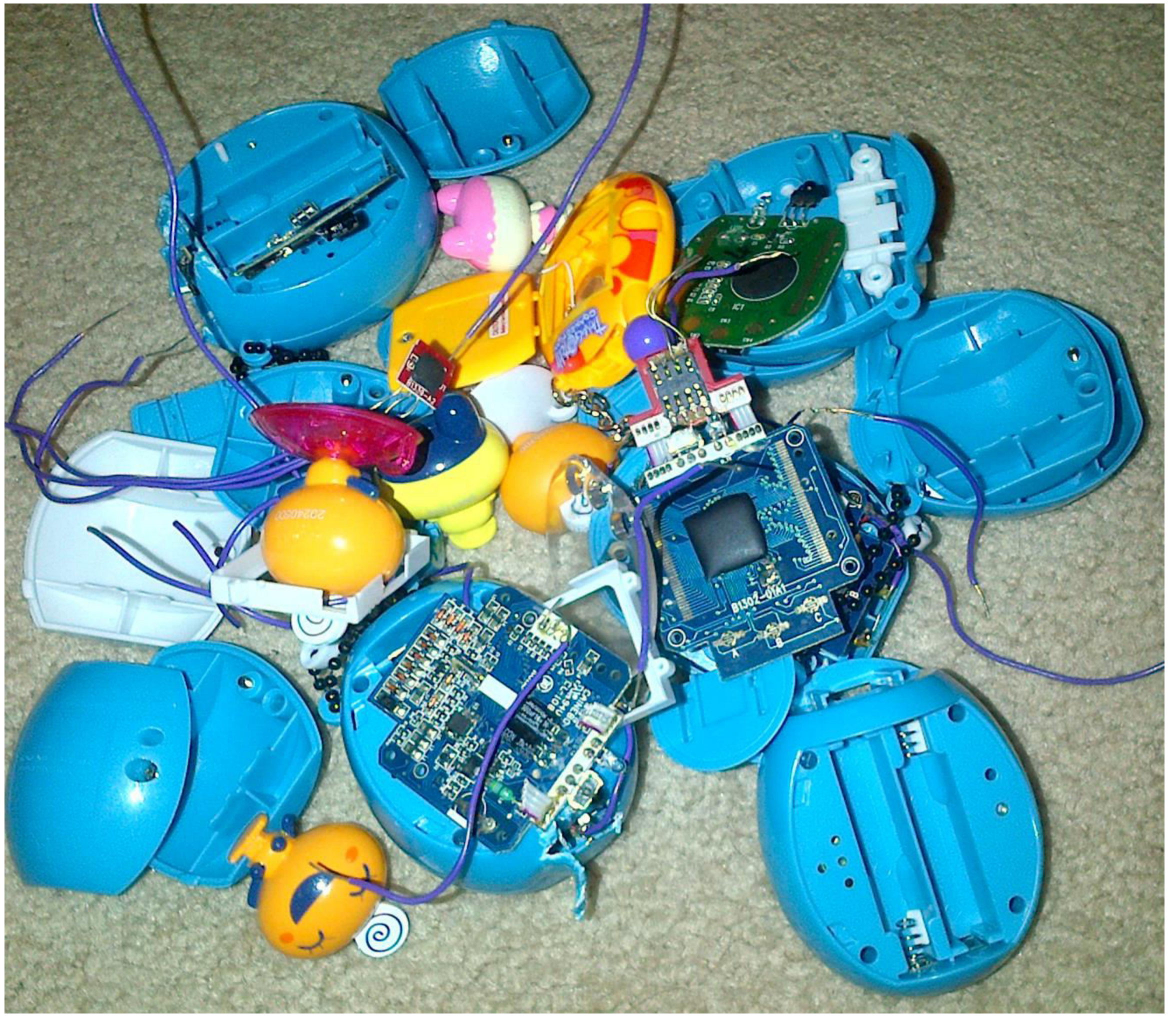
Buttons

Conclusion

Conclusions

- Dumped Tamagotchi code
- Learned about Tamagotchi internals
- Learned the secrets of Tamagotchi life
- Made Tamagotchis do new things
- Most importantly, good times were had by all…

Except for the Tamagotchis



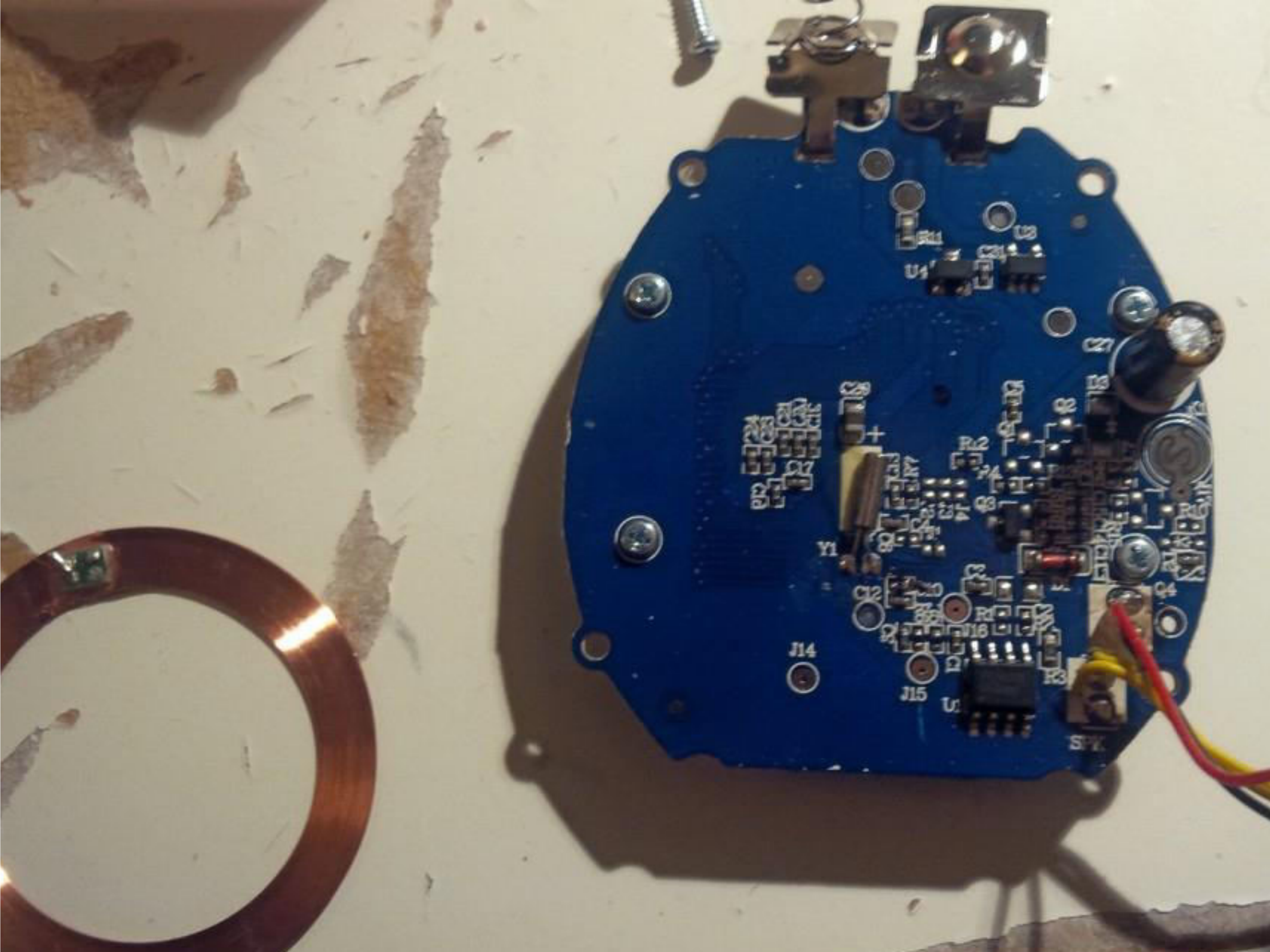
Tamagotchi Friends

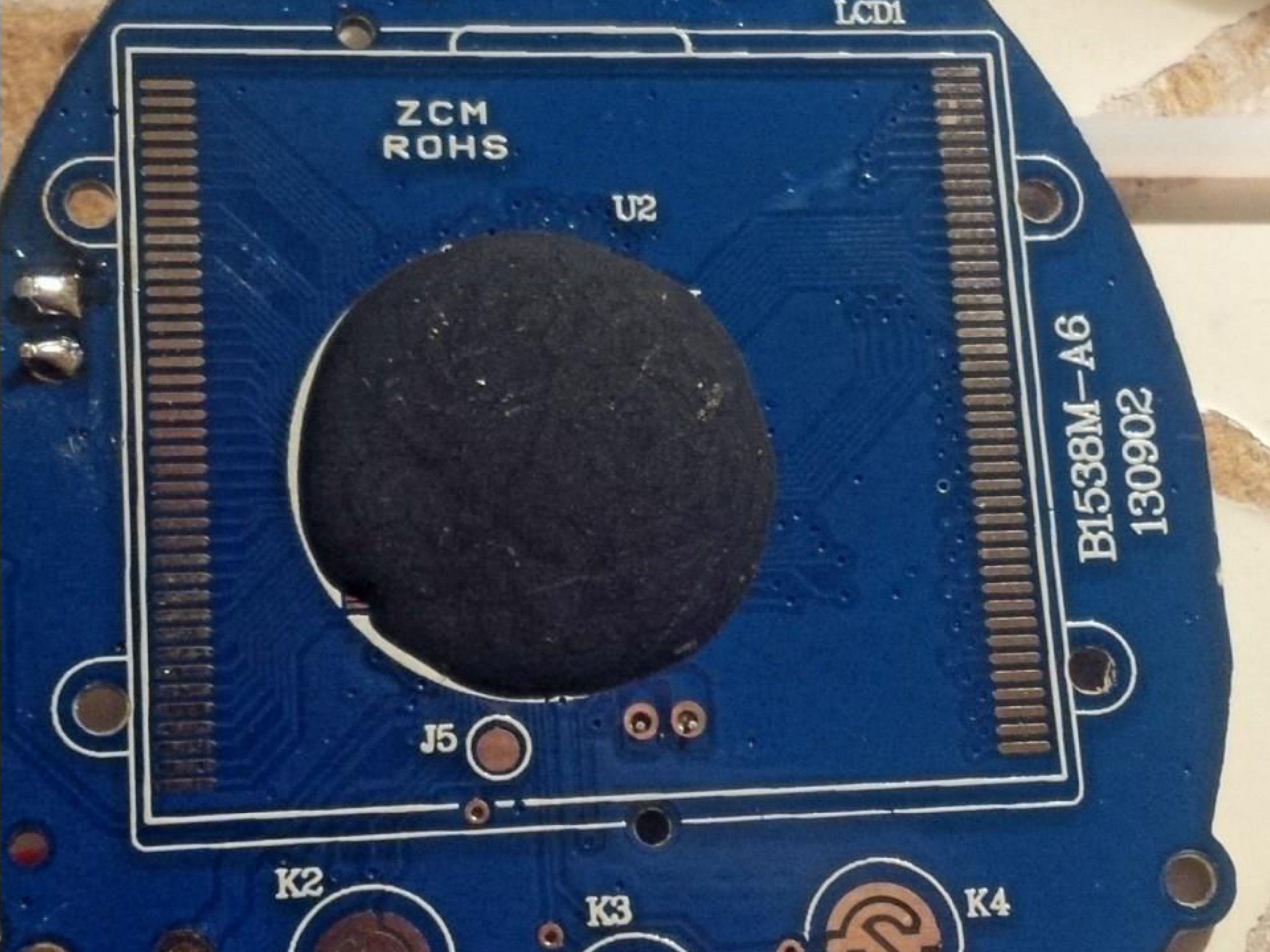
A New Tamagotchi!

Tamagotchi friends

Tamagotchi Friends

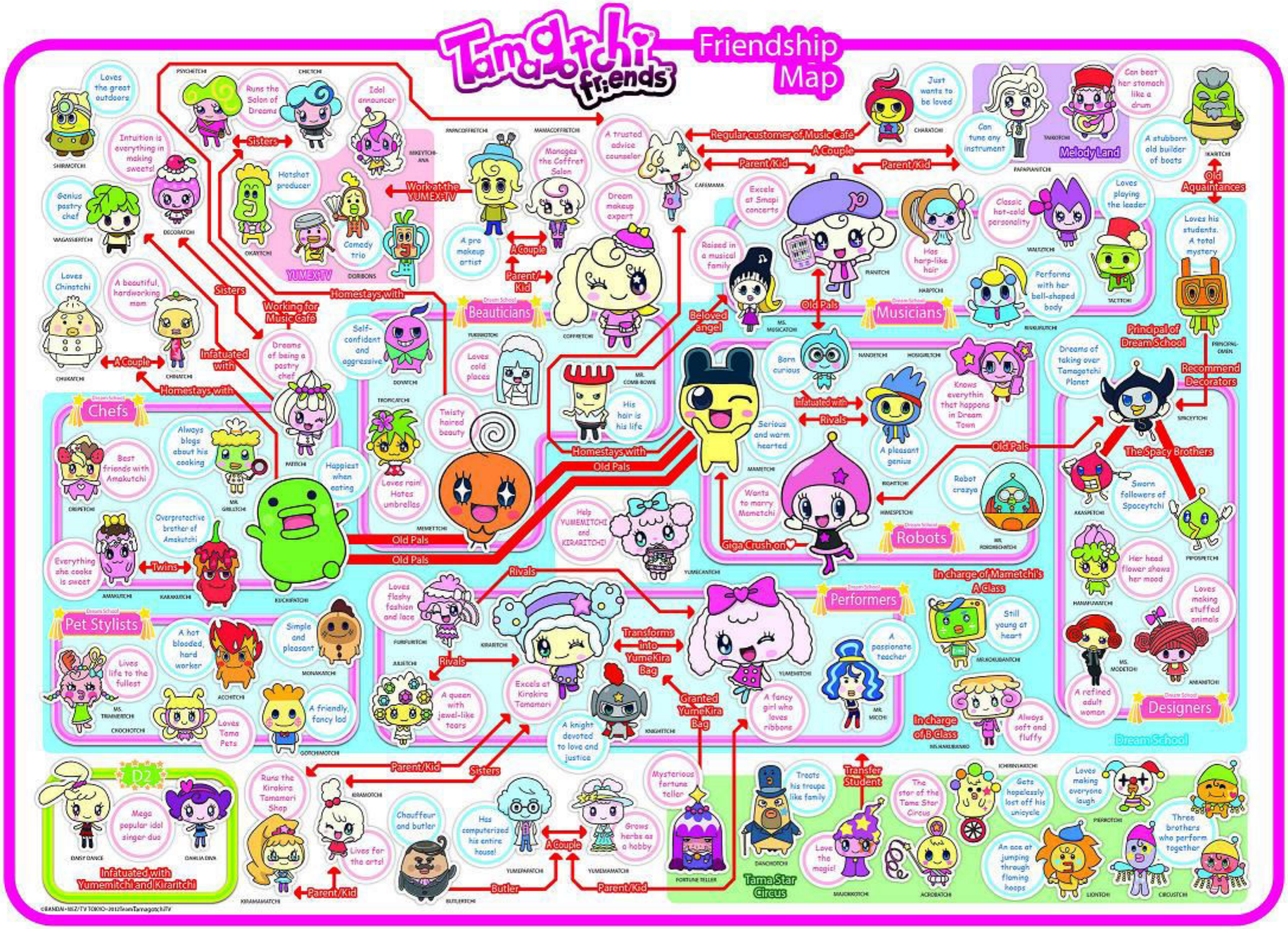
- Similar LCD and form factor
  - No IR or figures
  - Contains NFC
    - Send gifts
    - Visit
    - Send messages
    - Daily limits


U1
U3
C27
Y1
J14
J15
Q4
SPK


LCD1
ZCM
ROHS
U2
B1538M-A6
130902
J5
K2
K3
K4

Is it Hackable?

- Tamagotchi Friends probably uses the same MCU as the Tama-Go
  - Same form factor and LCD
- If it does, code can be dumped using the GeneralPlus test program
  - Decapping may be required
  - Reduced attack surface for code execution
- If not, who knows?



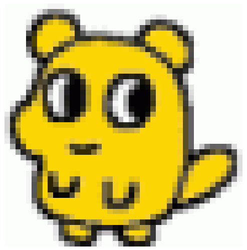
Tamagotchi friends
Friendship Map
Loves the great outdoors
SHIRAMOTCHI
Intuition is everything in making sweets!
DECORATCHI
Genius pastry chef
WAGASHIRETCHI
Loves Chinatchi
A beautiful, hardworking mom
CHUKATCHI
A Couple
CHINATCHI
PSYCHETCHI
Runs the Salon of Dreams
CHICTCHI
Sisters
Idol announcer
MIKEYTCHI-ANA
Hotshot producer
OKAYTCHI
Comedy trio
DORIBONS
YUMEX-TV
Work at the YUMEX-TV
PAPACOFFRETCHI
MAMACOFFRETCHI
Manages the Coffret Salon
A pro makeup artist
A Couple
Parent/Kid
Dream makeup expert
COFFRETCHI
A trusted advice counselor
CAFEMAMA
Regular customer of Music Café
A Couple
Parent/Kid
Parent/Kid
Just wants to be loved
CHARATCHI
Can tune any instrument
PAPAPIANITCHI
TAKTCHI
Can beat her stomach like a drum
Melody Land
A stubborn old builder of boats
IKARITCHI
Old Aquaintances
Sisters
Working for Music Café
Homestays with
Infatuated with
Dreams of being a pastry chef
Homestays with
MEITCHI
Beauticians
Self-confident and aggressive
DOYATCHI
Loves cold places
TUNDRATCHI
TROPICATCHI
Twisty haired beauty
MEMETCHI
Loves rain! Hates umbrellas
MR. COMB-BOWIE
His hair is his life
Beloved angel
Homestays with
Old Pals
Raised in a musical family
MS. MUSICATCHI
Excels at Smapi concerts
PIANITCHI
Has harp-like hair
HARPTCHI
Classic hot-cold personality
VAITSITCHI
Loves playing the leader
TACTTCHI
Performs with her bell-shaped body
RINKURUTCHI
Musicians
Old Pals
Loves his students. A total mystery
PRINCIPAL-DMEN
Principal of Dream School
Recommend Decorators
Born curious
NANDETCHI
Infatuated with
Rivals
HOSIGIRLTCHI
Knows everythin that happens in Dream Town
Dreams of taking over Tamagotchi Planet
SPACEYTCHI
The Spacy Brothers
Chefs
Always blogs about his cooking
MR. GRILLTCHI
Best friends with Amakutchi
CREPETCHI
Happiest when eating
KUCHIPATCHI
Old Pals
Old Pals
Overprotective brother of Amakutchi
Everything she cooks is sweet
Twins
AMAKUTCHI
KARAKUTCHI
Help YUMEMITCHI and KIRARITCHI
YUMECANTCHI
Serious and warm hearted
MAMETCHI
A pleasant genius
KIKITCHI
Wants to marry Mametchi
HIMESPETCHI
Giga Crush on ♥
Robot crazy
MR. ROBOBCHATCHI
Robots
Old Pals
AKASPETCHI
Sworn followers of Spaceytchi
PIPOSPETCHI
Her head flower shows her mood
HANAFUWATCHI
Loves making stuffed animals
Pet Stylists
Lives life to the fullest
MS. TRAINERTCHI
A hot blooded, hard worker
ACCHITCHI
Simple and pleasant
MONAKATCHI
Loves Tama Pets
CHOCHOTCHI
A friendly, fancy lad
GOTCHMOTCHI
Loves flashy fashion and lace
FURIFURITCHI
Rivals
KIRARITCHI
Rivals
JULIETCHI
A queen with jewel-like tears
Excels at Kirakira Tamamori
A knight devoted to love and justice
KNIGHTCHI
Transforms into YumeKira Bag
Granted YumeKira Bag
YUMEMITCHI
A fancy girl who loves ribbons
Performers
A passionate teacher
MR. MICCHI
In charge of Mametchi's A Class
Still young at heart
MR. KOKUBANTCHI
In charge of B Class
Always soft and fluffy
MS. MAKUBANKO
A refined adult woman
MS. MODETCHI
ANIANITCHI
Designers
Dream School
D2
Mega popular idol singer duo
DAISY DANCE
DAHLIA DIVA
Infatuated with Yumemitchi and Kiraritchi
Parent/Kid
Sisters
Runs the Kirakira Tamamori Shop
KIRAMOTCHI
Lives for the arts!
KIRAMAMATCHI
Parent/Kid
Chauffeur and butler
BUTLERTCHI
Butler
Has computerized his entire house!
YUMEPAPATCHI
A Couple
YUMEMAMATCHI
Grows herbs as a hobby
Parent/Kid
Mysterious fortune teller
FORTUNE TELLER
Treats his troupe like family
DANCHOKOTCHI
Transfer Student
Love the magic!
MAJOKKOTCHI
The star of the Tama Star Circus
ACROBATCHI
ICHIRINSHATCHI
Gets hopelessly lost off his unicycle
Loves making everyone laugh
PIERROTCHI
Three brothers who perform together
An ace at jumping through flaming hoops
LIONTCHI
CIRCUSTCHI
Tama Star Circus
©BANDAI·WiZ/TV TOKYO·2012TeamTamagotchiTV

Questions?

natalie@natashenka.ca

@natashenka

More Info



http://natashenka.ca

natalie@natashenka.ca

@natashenka