

ILFA6502 v2.1.7-refresh5 チュートリアル・ファーストステップガイド

ファミコンソフト開発初心者向け / Windows・Linux 対応

2026-07-08

Contents

ILFA6502 v2.1.7-refresh5 チュートリアル・ファーストステップガイド	1
1. このガイドの目的	1
2. まず覚える考え方	2
3. Windows 版の前提	2
4. Linux 版の前提	2
5. 最初の確認: menu と doctor	3
6. テンプレート一覧を見る	3
7. チュートリアル 1: hello ROM を作る	4
8. エミュレータで開く	5
9. プロジェクトの中身を見る	5
10. main.asm を読む	5
11. チュートリアル 2: リバーシを作る	6
12. リバーシのファイル構成	7
13. 既存プロジェクトを作り直す	7
14. サンプル一覧を見る	8
15. template-info を使う	8
16. SDK を調べる	8
17. ファミコン開発の最小知識	9
18. 最初の改造: プロジェクト名を変えて作る	10
19. 最初の改造: main.asm を編集する	10
20. .lst、.map、.sym の見方	10
21. 警告の読み方	10
22. よくある失敗と対処	11
23. 推奨学習順	12
24. 次に読むヘルプ	12
25. 最小コマンド集	12
26. まとめ	13

ILFA6502 v2.1.7-refresh5 チュートリアル・ファーストステップガイド

1. このガイドの目的

このガイドは、ファミコンソフト開発がはじめての利用者が、ILFA6502 v2.1.7-refresh5 を使って、最初の .nes ROM を作成し、確認し、少し改造してみるところまでを案内するための入門書です。

本書では、6502 CPU、PPU、APU、NROM などの詳細仕様を完全に理解していなくても、まず手を動かして ROM を生成できることを優先します。

目標は次の通りです。

1. ILFA6502 が起動することを確認する。
2. 内蔵テンプレートから hello プロジェクトを作る。
3. build で .nes ROM を生成する。
4. verify-rom で ROM の形式を確認する。
5. FCEUX などのエミュレータで ROM を開く。
6. テンプレートやサンプルの一覧を見て、次に試す題材を選ぶ。
7. main.asm、ilfa_project.toml、assets、build の役割を理解する。

2. まず覚える考え方

ILFA6502 の基本的な流れは、次の 3 段階です。

```
new      : テンプレートからプロジェクトを作る
build    : プロジェクトをビルドして .nes ROM を作る
verify   : 作成した .nes ROM を検査する
```

重要なのは、new だけでは ROM は作られない、という点です。

new はフォルダとソースを作るだけです。

build を実行して、はじめて .nes ROM が生成されます。

この順序を間違えると、verify-rom 実行時に ROM ファイルが見つからない、というエラーになります。

3. Windows 版の前提

Windows ビルドキットで生成した EXE を使う場合、作業はビルドキットの親フォルダで行います。

推奨する呼び出し方は次の形式です。

```
dist\ilfa6502.exe menu
dist\ilfa6502.exe doctor
dist\ilfa6502.exe list templates --long
```

dist フォルダの中へ移動して作業することもできますが、プロジェクトの作成場所が分かりにくくなります。初心者の間は、親フォルダから dist\ilfa6502.exe を呼び出す方法を推奨します。

3.1 Windows でよくある入力ミス

Windows のコマンドプロンプトでは、パス区切りに \ を使います。

正しい例:

```
dist\ilfa6502.exe template-info reversi_famicom
```

間違いやすい例:

```
dist/ilfa6502.exe template-info reversi_famicom
/dist/ilfa6502.exe template-info reversi_famicom
```

また、PDF や文書からコピーしたときに、--target のハイフンが見ただけ似た別文字になることがあります。refresh5 では特殊ハイフン補正が入っていますが、コマンド例では必ず ASCII の - を使ってください。

安全な例:

```
dist\ilfa6502.exe new hello --target nes-nrom --template hello
```

4. Linux 版の前提

Linux 版では、実行ファイルに実行権限を付与してから使います。

```
chmod +x ./ilfa6502
./ilfa6502 menu
./ilfa6502 doctor
./ilfa6502 list templates --long
```

Linux ビルドキットで生成した場合は、次のように実行します。

```
chmod +x ./dist/ilfa6502
./dist/ilfa6502 menu
./dist/ilfa6502 doctor
./dist/ilfa6502 list templates --long
```

5. 最初の確認: menu と doctor

最初に menu を表示します。

Windows:

```
dist\ilfa6502.exe menu
```

Linux:

```
./ilfa6502 menu
```

menu は、よく使うコマンドの入口です。

次に doctor を実行します。

Windows:

```
dist\ilfa6502.exe doctor
```

Linux:

```
./ilfa6502 doctor
```

doctor では、テンプレート、SDK、サンプルなどの内蔵リソースが見つかるかを確認できます。

正常例の目安:

```
ILFA6502 doctor
version      : v2.1.7-refresh5
templates    : 19
sdk modules  : 20
examples     : 19
status       : OK
```

status : OK が出ていれば、基本的な実行環境は整っています。

6. テンプレート一覧を見る

どのようなプロジェクトを作れるかを見るには、テンプレート一覧を表示します。

Windows:

```
dist\ilfa6502.exe list templates --long
```

Linux:

```
./ilfa6502 list templates --long
```

代表的なテンプレートは次の通りです。

テンプレート	内容
hello	最小構成の NROM 起動確認用。最初に試す題材。
ppu	PPU 初期化、パレット、文字表示の確認用。
sprite	スプライト表示の基本サンプル。
pad	コントローラ入力の基本サンプル。
mml_sound	簡易 MML / APU 音楽再生の確認用。
block_breaker	ブロック崩し。ゲームらしい動作確認用。
galaxy_shooter	シューティングゲーム型のサンプル。

テンプレート	内容
reversi_famicom	v2.1.7 代表サンプル。ファミコン版リバーシ。

最初は hello、次に ppu、sprite、pad、最後に reversi_famicom へ進むと理解しやすいです。

7. チュートリアル 1: hello ROM を作る

ここから実際に ROM を作ります。

7.1 Windows ビルドキット版

作業フォルダで次を実行します。

```
dist\ilfa6502.exe new hello --target nes-nrom --template hello
```

成功すると、次のような表示が出ます。

```
OK: created hello
```

```
Next:
```

```
Recommended from the current folder:
```

```
"...\dist\ilfa6502.exe" build "hello" --listing --map --symbols --warnings
```

続けてビルドします。

```
dist\ilfa6502.exe build hello --listing --map --symbols --warnings
```

生成された ROM を探します。

```
dir /s /b hello\*.nes
```

通常は次の場所にできます。

```
hello\build\hello.nes
```

ROM を検査します。

```
dist\ilfa6502.exe verify-rom hello\build\hello.nes
```

7.2 Linux 版

```
./ilfa6502 new hello --target nes-nrom --template hello
```

```
./ilfa6502 build hello --listing --map --symbols --warnings
```

```
find hello -name "*.nes" -print
```

```
./ilfa6502 verify-rom hello/build/hello.nes
```

7.3 verify-rom の見方

verify-rom では、iNES ヘッダ、PRG ROM、CHR ROM、Mapper、サイズ、ベクタを確認します。

正常例:

```
iNES ROM check
```

```
PRG ROM : 2 x 16KB = 32768 bytes
```

```
CHR ROM : 1 x 8KB = 8192 bytes
```

```
mapper : 0
```

```
size chk : OK
```

```
vectors :
```

```
RESET = $8000
```

```
result : OK: header, size and vectors look valid
```

result : OK が出れば、ROM 形式としては基本的に正常です。

ただし、verify-rom はゲーム内容の正しさまでは保証しません。画面表示や操作はエミュレータで確認します。

8. エミュレータで開く

生成した .nes ファイルは、FCEUX などの NES / ファミコン用エミュレータで開きます。

Windows の例:

```
fceux hello\build\hello.nes
```

FCEUX にパスが通っていない場合は、FCEUX の画面から File、Open ROM を選び、生成された .nes ファイルを開いてください。

hello は最小構成なので、画面上の変化は少ない場合があります。まずは「ROM として読み込める」ことを確認する題材です。

9. プロジェクトの中身を見る

hello プロジェクトを作ると、主に次のファイルができます。

```
hello/  
  ilfa_project.toml  
  main.asm  
  build/  
    hello.nes  
    hello.lst  
    hello.map  
    hello.sym
```

ファイル	役割
ilfa_project.toml	ビルド設定ファイル。ソース、出力先、include pathなどを指定します。
main.asm	メインの 6502 アセンブリソースです。
build/*.nes	生成された NES ROM です。
build/*.lst	アセンブル結果のリストです。デバッグ時に使います。
build/*.map	ラベルや配置の確認に使います。
build/*.sym	シンボル情報です。

初心者の間は、まず main.asm と ilfa_project.toml の存在を確認できれば十分です。

10. main.asm を読む

hello の main.asm は、最小構成の NES ROM を作るための見本です。

典型的には、次のような構成になっています。

```
.include "nes/nes.inc"
```

```
NES_HEADER 2, 1, 0, 0  
NES_PRG_BEGIN
```

```
RESET:  
    NES_INIT  
    WAIT_VBLANK  
    PPU_DISABLE
```

```
main:  
    JMP main
```

```
NMI:  
    RTI
```

IRQ:
RTI

NES_VECTORS NMI, RESET, IRQ
NES_CHR_EMPTY

各部分の意味は次の通りです。

部分	意味
.include "nes/nes.inc"	NES 向けの基本マクロを読み込みます。
NES_HEADER	iNES ヘッダを生成します。
NES_PRG_BEGIN	PRG ROM 領域の開始を示します。
RESET:	電源投入時、リセット時に実行される入口です。
NES_INIT	NES の基本初期化を行うマクロです。
WAIT_VBLANK	VBlank を待ちます。
PPU_DISABLE	PPU 描画を無効にします。
NMI:	NMI 割り込み処理です。
IRQ:	IRQ 割り込み処理です。
NES_VECTORS	NMI / RESET / IRQ ベクタを ROM 末尾に配置します。
NES_CHR_EMPTY	空の CHR ROM を付けます。

ここで大事なのは、ファミコン ROM には RESET の入口とベクタが必要だということです。verify-rom の RESET = \$8000 のような表示は、この起動位置を確認するためのものです。

11. チュートリアル 2: リバースを作る

次に、v2.1.7 の代表サンプルである reversi_famicom を作ります。

Windows:

```
dist\ilfa6502.exe new testReversi --target nes-nrom --template reversi_famicom
dist\ilfa6502.exe build testReversi --listing --map --symbols --warnings
dist\ilfa6502.exe verify-rom testReversi\build\testReversi.nes
```

Linux:

```
./ilfa6502 new testReversi --target nes-nrom --template reversi_famicom
./ilfa6502 build testReversi --listing --map --symbols --warnings
./ilfa6502 verify-rom testReversi/build/testReversi.nes
```

正常なら、verify-rom は次のような内容を表示します。

```
PRG ROM   : 2 x 16KB = 32768 bytes
CHR ROM   : 1 x 8KB  = 8192 bytes
mapper    : 0
size chk  : OK
vectors   :
  NMI     = $8B92
  RESET   = $8000
  IRQ     = $8BB7
result    : OK: header, size and vectors look valid
```

生成された ROM は次の場所にあります。

Windows:

testReversi\build\testReversi.nes

Linux:

testReversi/build/testReversi.nes

12. リバーシのファイル構成

reversi_famicom では、hello より多くのファイルが作られます。

```
testReversi/  
  ilfa_project.toml  
  main.asm  
  assets/  
    bg/  
    chr/  
    mml/  
  build/  
    testReversi.nes  
    testReversi.lst  
    testReversi.map  
    testReversi.sym
```

フォルダ	内容
assets/bg	タイトル画面、盤面、結果画面などの背景データです。
assets/chr	キャラクタパターン、タイル画像データです。
assets/mml	BGM や効果音の元になる MML データです。
build	生成物の出力先です。

リバーシのようなゲーム系テンプレートは、MML 変換や include path の設定が必要です。そのため、直接 asm main.asm を実行せず、必ず build を使ってください。

正しい例:

```
dist\ilfa6502.exe build testReversi --listing --map --symbols --warnings
```

避ける例:

```
dist\ilfa6502.exe asm testReversi\main.asm -o testReversi.nes --target nes-nrom
```

asm は単体 ASM ファイル向けです。ゲームテンプレート全体の構築には build を使います。

13. 既存プロジェクトを作り直す

同じ名前のフォルダがすでにある場合、new は失敗します。

例:

```
ERROR: destination already exists and is not empty: testReversi
```

作り直す場合は、先に削除します。

Windows:

```
rmdir /s /q testReversi  
dist\ilfa6502.exe new testReversi --target nes-nrom --template reversi_famicom
```

Linux:

```
rm -rf testReversi  
./ilfa6502 new testReversi --target nes-nrom --template reversi_famicom
```

消したくない場合は、別名で作ります。

```
dist\ilfa6502.exe new testReversi2 --target nes-nrom --template reversi_famicom
```

14. サンプル一覧を見る

サンプル一覧は次のコマンドで確認します。

Windows:

```
dist\ilfa6502.exe list samples --long
```

Linux:

```
./ilfa6502 list samples --long
```

samples は学習用の完成例です。templates は new コマンドで複製する雛形です。

区分	目的
templates	新規プロジェクトを作るための雛形。
samples	読む、学ぶ、動作確認するためのサンプル。
sdk	.include で使う共通ライブラリ。

迷ったら、まず `list templates --long` で雛形を選び、`new` でプロジェクト化してください。

15. template-info を使う

テンプレートに含まれるファイルを確認するには、`template-info` を使います。

Windows:

```
dist\ilfa6502.exe template-info reversi_famicom
```

Linux:

```
./ilfa6502 template-info reversi_famicom
```

表示例:

```
TEMPLATE: reversi_famicom
description: v2.1.7 ファミコン版リバーシ。
project file: templates\reversi_famicom\ilfa_project.toml
output      : build/{{PROJECT_NAME}}.nes
files:
- assets\bg\board.atr
- assets\bg\board.nam
- assets\chr\reversi.chr
- assets\mml\reversi_theme.mml
- ilfa_project.toml
- main.asm
```

大きなテンプレートを使う前に、何が含まれているかを確認するのに便利です。

16. SDK を調べる

SDK の一覧は次のコマンドで確認します。

Windows:

```
dist\ilfa6502.exe list sdk --long --deps
```

Linux:

```
./ilfa6502 list sdk --long --deps
```

特定の SDK モジュールを調べるには、`sdk-info` を使います。

```
dist\ilfa6502.exe sdk-info nes/game.inc
```

代表的な SDK ファイルは次の通りです。

SDK	役割
nes/nes.inc	NES 向け基本定義と基本マクロ。
nes/nrom.inc	NROM 構成支援。
nes/ppu.inc	PPU レジスタや描画支援。
nes/pad.inc	コントローラ入力支援。
nes/apu.inc	APU 音源支援。
nes/sfx.inc	効果音支援。
nes/vectors.inc	NMI / RESET / IRQ ベクタ支援。
nes/game.inc	ゲーム系テンプレートの共通支援。

17. ファミコン開発の最小知識

ここでは、最初に知っておくと迷いにくい用語だけを説明します。

17.1 6502 CPU

ファミコンの中心となる CPU です。ILFA6502 は、この CPU 向けのアセンブリ言語を機械語に変換します。

17.2 PRG ROM

プログラムを入れる ROM 領域です。6502 の命令やデータが入ります。

17.3 CHR ROM

キャラクタパターン、つまりタイル画像を入れる ROM 領域です。ファミコンの背景やスプライトの絵柄に使います。

17.4 PPU

画面表示を担当するチップです。背景、スプライト、パレットなどを扱います。

17.5 APU

音を担当するチップです。BGM や効果音を鳴らします。

17.6 NMI

画面更新タイミングで発生する割り込みです。ファミコンでは、画面更新処理の入口として重要です。

17.7 RESET ベクタ

電源投入やリセット時に CPU が最初に実行するアドレスを示す情報です。verify-rom で確認できます。

17.8 NROM

もっとも基本的な NES ROM 形式です。ILFA6502 の初心者向けテンプレートは、主に NROM を対象にしています。

18. 最初の改造: プロジェクト名を変えて作る

同じテンプレートから、別名のプロジェクトを作ることができます。

```
dist\ilfa6502.exe new myHello --target nes-nrom --template hello
dist\ilfa6502.exe build myHello --listing --map --symbols --warnings
dist\ilfa6502.exe verify-rom myHello\build\myHello.nes
```

名前を変えるだけでも、出力される ROM 名が変わります。

myHello\build\myHello.nes

19. 最初の改造: main.asm を編集する

hello は画面表示をほとんどしない最小構成です。画面表示を試したい場合は、text や ppu テンプレートのほうが向いています。

```
dist\ilfa6502.exe new textTest --target nes-nrom --template text
dist\ilfa6502.exe build textTest --listing --map --symbols --warnings
dist\ilfa6502.exe verify-rom textTest\build\textTest.nes
```

textTest\main.asm を開くと、文字表示用の処理や文字列データが確認できます。まずは文字列らしき部分を探し、短い英数字に変更して再ビルドしてみてください。

手順:

```
dist\ilfa6502.exe build textTest --listing --map --symbols --warnings
```

編集後に必ずビルドし直してください。ソースを保存しただけでは ROM は更新されません。

20. .lst、.map、.sym の見方

--listing --map --symbols --warnings を付けてビルドすると、ROM と一緒に補助ファイルが作られます。

ファイル	使い方
.lst	アセンブルされた命令とアドレスを確認します。
.map	セグメントやラベルの配置を確認します。
.sym	ラベル名とアドレスを確認します。

初心者のうちは、エラーが出たときに .lst と .map を見る、という程度で十分です。

21. 警告の読み方

ビルド時に次のような警告が出ることがあります。

WARNINGS:

- multiple output segments detected; default binary concatenates segments in declaration order
- unused label: tmp0 (main.asm:53)

unused label は、定義したラベルが使われていないという意味です。多くの場合、すぐに致命的ではありません。

multiple output segments detected は、複数の出力セグメントがあるため、宣言順に連結するという警告です。verify-rom が OK で、エミュレータでも動作するなら、最初の段階では様子見で構いません。

ただし、ROM が暴走する、画面が出ない、音が止まらないなどの場合は、.map と verify-rom の結果を確認してください。

22. よくある失敗と対処

22.1 ROM file not found

例:

```
ERROR: ROM file not found: testReversi\build\testReversi.nes
```

原因は、多くの場合、まだ build を実行していないことです。

対処:

```
dist\ilfa6502.exe build testReversi --listing --map --symbols --warnings  
dist\ilfa6502.exe verify-rom testReversi\build\testReversi.nes
```

22.2 destination already exists

例:

```
ERROR: destination already exists and is not empty: testReversi
```

同じ名前のフォルダがすでにあります。

対処 1: 削除して作り直す。

```
rmdir /s /q testReversi  
dist\ilfa6502.exe new testReversi --target nes-nrom --template reversi_famicom
```

対処 2: 別名で作る。

```
dist\ilfa6502.exe new testReversi2 --target nes-nrom --template reversi_famicom
```

22.3 ilfa6502.exe が見つからない

Windows ビルドキットでは、実行ファイルは dist\ilfa6502.exe にあります。

正しい例:

```
dist\ilfa6502.exe menu
```

親フォルダで単に次のように打つと失敗します。

```
ilfa6502.exe menu
```

22.4 dist が認識されない

Windows で次のように / を使うと失敗する場合があります。

```
dist/ilfa6502.exe menu
```

Windows では次のように \ を使ってください。

```
dist\ilfa6502.exe menu
```

22.5 PDF からコピーしたコマンドが動かない

--target や template-info の - が特殊ハイフンになっている可能性があります。

refresh5 では補正機能がありますが、確実にするには COMMANDS_REFRESH5.txt のようなプレーンテキスト版からコピーしてください。

23. 推奨学習順

初心者は、次の順に試すと理解しやすいです。

1. hello
2. ppu
3. text
4. sprite
5. pad
6. sound
7. mml_sound
8. scroll
9. block_breaker
10. galaxy_shooter
11. reversi_famicom

最初から reversi_famicom を読むと、量が多くて大変です。まずは hello、text、pad のような小さいテンプレートで慣れることを推奨します。

24. 次に読むヘルプ

ILFA6502 には、目的別のヘルプがあります。

```
dist\ilfa6502.exe help quickstart
dist\ilfa6502.exe help commands
dist\ilfa6502.exe help workflow
dist\ilfa6502.exe help options
dist\ilfa6502.exe help templates
dist\ilfa6502.exe help sdk
dist\ilfa6502.exe help nes
dist\ilfa6502.exe help ppu
dist\ilfa6502.exe help mml
```

迷ったら、まず次の3つです。

```
dist\ilfa6502.exe menu
dist\ilfa6502.exe help quickstart
dist\ilfa6502.exe list templates --long
```

25. 最小コマンド集

Windows ビルドキット版の最小手順です。

```
dist\ilfa6502.exe doctor
dist\ilfa6502.exe list templates --long
dist\ilfa6502.exe new hello --target nes-nrom --template hello
dist\ilfa6502.exe build hello --listing --map --symbols --warnings
dist\ilfa6502.exe verify-rom hello\build\hello.nes
```

リバーシ版です。

```
dist\ilfa6502.exe new testReversi --target nes-nrom --template reversi_famicom
dist\ilfa6502.exe build testReversi --listing --map --symbols --warnings
dist\ilfa6502.exe verify-rom testReversi\build\testReversi.nes
```

Linux 版の最小手順です。

```
./ilfa6502 doctor
./ilfa6502 list templates --long
./ilfa6502 new hello --target nes-nrom --template hello
./ilfa6502 build hello --listing --map --symbols --warnings
./ilfa6502 verify-rom hello/build/hello.nes
```

26. まとめ

ILFA6502 の最初の使い方は、難しく考えすぎないことが大切です。

まずは、次の 3 つを覚えてください。

new : プロジェクトを作る
build : ROM を作る
verify-rom : ROM の形式を確認する

そして、困ったときは次を実行します。

```
dist\ilfa6502.exe menu  
dist\ilfa6502.exe help quickstart  
dist\ilfa6502.exe list templates --long
```

ファミコン開発は、CPU、PPU、APU、CHR、NMI、ベクタなど、最初は見慣れない用語が多く出てきます。しかし、最初からすべてを理解する必要はありません。

まず ROM を作る。次に動かす。少し変更して、またビルドする。この繰り返しで、少しずつ理解できます。

ILFA6502 v2.1.7-refresh5 は、その最初の一步を踏み出すための小さな開発キットです。