# Multi-Language Programming on the P2 with fastspin

Basic Overview

July 1, 2020

# FlexGUI vs fastspin

- Flexgui: simple interface to fastspin compiler
  - Basic editing and running functions, nothing particularly special
  - Could be replaced by SpinEdit, VS Code, or whatever
- The magic all happens in the command line tools: fastspin and loadp2.
- Fastspin is the compiler that turns BASIC, C, or Spin into P2 (or P1!) machine code

# Why fastspin for P2?

- Multi-language: C, BASIC, Spin1, Spin2
  - Spin2 because that's the official P2 language
  - Spin1 to help you port your code from P1
  - Incorporate C or BASIC code from 3rd parties
- Supports both P1 and P2
  - Can make same program run on both
    - (if you avoid hardware specific things like PASM, and/or use #ifdef)
- Cross platform
  - Windows, macOS, Linux

## Why fastspin? (part 2)

- Features
  - Built in preprocessor
  - Listing files
- Performance
  - Produces native code
  - Optimizations:
    - “a++”, “a += 1”, “a = a+1” all produce the same code
    - Write the way you want, let the compiler worry about producing good code

# Optimization Example

```c
// add an array of integers to another
void addarray(int N, int *a, int *b)
{
    int i;
    for (i = 0; i < N; i++) {
        a[i] = a[i] + b[i];
    }
}
```

```
_addarray
               cmps   arg01, #0 wcz
 if_be    jmp         #LR__0003
               mov    _var01, arg01
               rep    @LR__0002, _var01
LR__0001
               rdlong _var02, arg02
               rdlong _var03, arg03
               add    _var02, _var03
               wrlong _var02, arg02
               add    arg02, #4
               add    arg03, #4
LR__0002
LR__0003
_addarray_ret
          reta
```

## Why NOT fastspin?

- Not completely Spin2 compatible yet
  - Actively working on this still
- C compiler not completely standard compliant
  - Libraries are incomplete
  - No linker, must compile whole program at once
  - Lax about order of declarations (accepts non-standard C)
- Code is machine code rather than bytecode
  - So needs more memory, about twice as much

## Let’s get started

- Start up FlexGUI
  - Editor options menu (e.g. make font bigger)
  - Documentation menu
  - Specials for P2
  - Built in terminal
- “Hello world” in BASIC
- “Hello world” in Spin
  - Can use the C library for this!

# PASM Programming

- Preprocessor
  - #define for debugging and portability
  - #ifdef / #error for checking code
- Warnings for some common assembler mistakes
  - Missing # in jumps
  - Missing wcz in cmp
- Address relocation when mixed with high level languages

## Spin Programming

- Default is Spin1
  - Works for P2 too, as long as assembly is P2ASM rather than PASM
  - Can use this as a stepping stone for porting P1 to P2
- If file extension is “.spin2”, Spin2 compatible mode
- Many extensions; the fastspin dialects of Spin1 and Spin2 have a big overlap

## Pure PASM

- Normally pure assembly code is wrapped up in a .spin2 file with just CON and DAT sections
- You could also wrap it in C or BASIC
  - Or, more likely, put in assembly in the C or BASIC file which is intended to be run in another COG
- See pure_pasm.spin2 or pure_pasm.c
- Useful fastspin features for PASM
  - Preprocessor
  - Warnings

# C language support

- Most of C99 language implemented
- Some C++ features as extensions
  - Simple classes
  - References, default parameter values
- TODO:
  - Mostly libraries
  - 64 bit integers and doubles would be nice
  - Need some kind of linker or something

## C Language Demo

- Mandelbrot
- Main code is in C, video driver in Spin
- Inline assembly for performance

# BASIC programming

- Simple syntax (very easy to get going)
- Built in I/O, floating point, other nice features
- Mostly MS-BASIC compatible for portability
  - Even really old school code with line numbers accepted!
- Like Spin, can have PASM blocks
- e.g. see LED_interactive.bas

## Mixing BASIC, C, and Spin/Spin2

- Can call C from Spin, Spin from BASIC, Spin from C... any combination
- Turtle graphics demo
  - rogloh’s video driver
  - Turtle.c from web
  - My own BASIC glue code
- Note host file system I/O

## Thanks!

- Questions? (if we have time)