

Transferring data between R5 <==> Linux

1) How we share data between Linux and the R5

Memory sharing

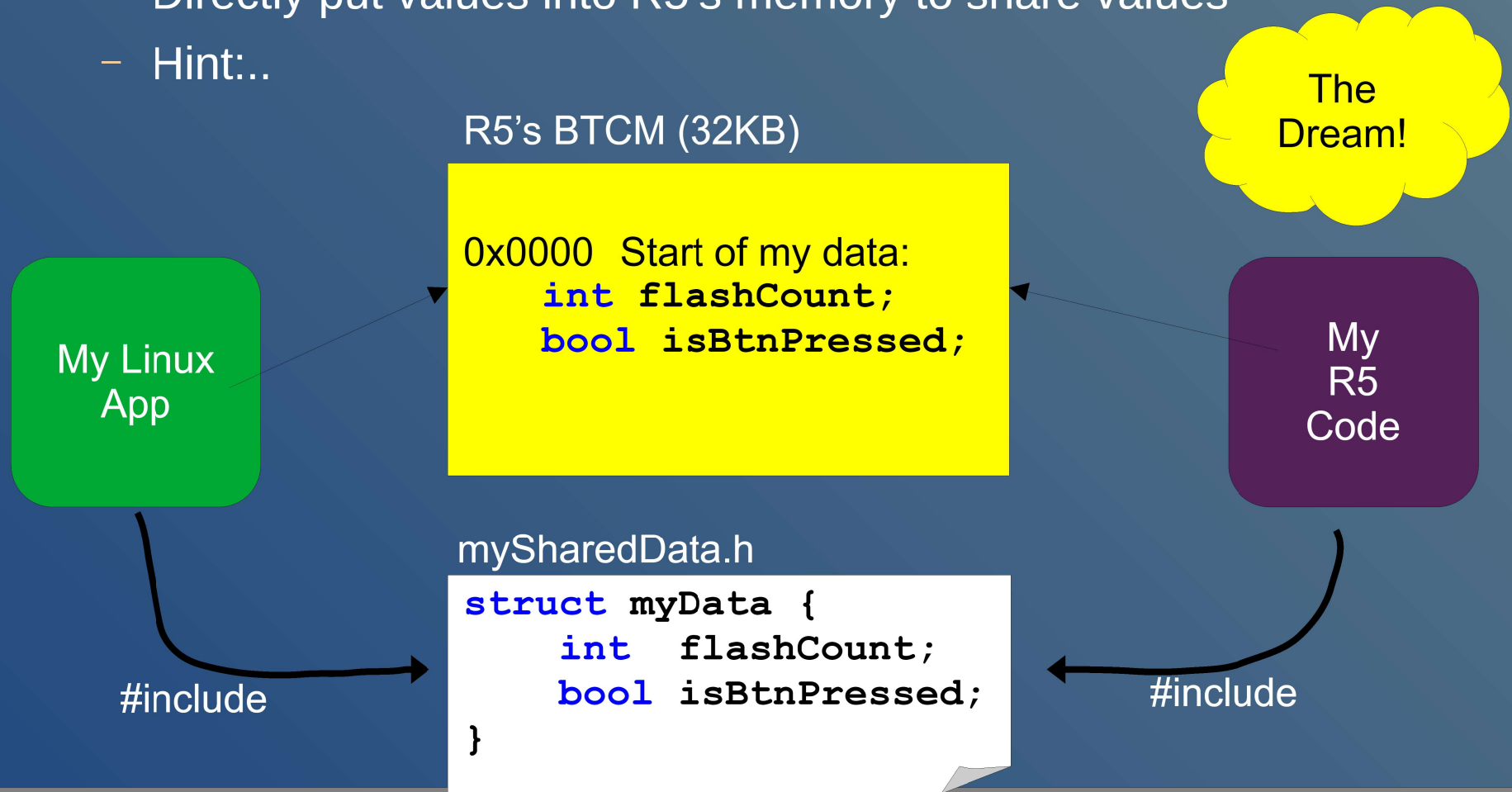
..	..	
0x7900 0000	R5 MCU ATCM (32KB)	0x0004 1010 (?)
0x7902 0000	R5 MCU BTCM (32KB, in 2 banks)	0x0000 0000

- To use BTCM, Linux global address 0x79020000
 - Must be mapped into your app's memory space with `mmap()`

Memory Use

- Shared Memory Idea

- Directly put values into R5's memory to share values
- Hint:..



Packing Structs

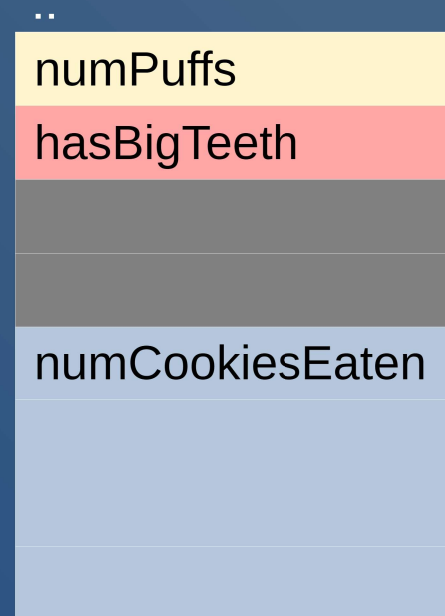
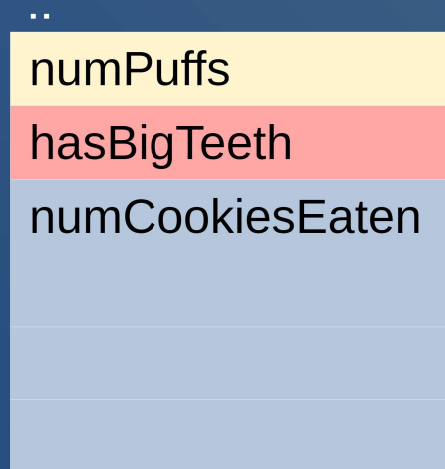
Data Types

- C data types can be of different sizes
 - C spec simply mentions their relative size
 - R5 and Linux use:
 - 1 byte: char
 - 2 bytes: short
 - 4 bytes: int, long, float
 - 8 bytes: long long, double
- ..
 - Gives integer data types based on #bits
 - Useful for..
 - uint8_t, uint16_t, uint32_t, uint64_t
 - int8_t, int16_t, int32_t, int64_t

Structs

- Structs store different types of data in one allocated unit of memory
- How does this layout in memory?

```
struct bigBadWolfData_t {  
    char numPuffs;  
    bool hasBigTeeth;  
    int numCookiesEaten;  
};
```



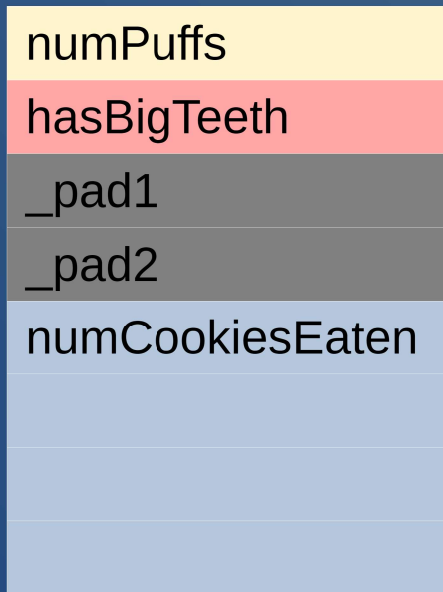
2 Processors

- Incorrect alignment gives a bus error
- Word align `int/uint32_t`
- Double word align `doubles, long long, uint64_t`

Padding Structs

```
struct bigBadWolfData_t {  
    char numPuffs;  
    bool hasBigTeeth;  
    char _pad1, _pad2;  
    int numCookiesEaten;  
};
```

Padded



Padding bytes

- Add extra bytes to `struct`..

`char/bool`: byte aligned

`int/uint32_t`: word aligned

`double/uint64_t`: dword aligned

- Once padded correctly, `struct` is identical as both packed and unpacked processors
 - Incorrect padding means values written to a field by one processor not seen correctly by other.

Troubleshooting

- Hard to debug the R5 because
 - ..
 - Write very little code at a time, then test it.
 - Flash the LED for some visual status
- Common Issues
 - Permission denied on /dev/mem:
run with sudo
 - Input/output not working:
check you have run GPIO code on Linux first
 - Data exchange problems:
R5 halts on struct access; use array.
 - Changes to code not running:
add compile-time error to check if correct code is compiling

Summary

- R5 Memory
 - 32KB in ATCM and BTCM banks
 - ~~Can use a struct to define which values are in shared memory~~
 - NOPE! Use raw memory / array
- Linux <==> R5 Memory
 - Linux app calls `mmap()` to request access to R5 memory
- Alignment / Packing
 - pad `structs` to line up data