

s

t

r

i

n

g

\0

0x100

0x101

0x102

0x103

0x104

0x105

0x106

Strings, sprintf & Buffer Arrays

Embedded C on the ATmega328P with MPLAB X IDE
Written with the help of Claude.ai

2 KB of SRAM. Every byte counts.

1

Today's roadmap

1

Strings in memory

char arrays, the '\0' terminator and the 2 KB SRAM budget

2

The string.h toolkit

Copy, join, compare and search without overrunning a buffer

3

sprintf & formatting

Format specifiers, AVR quirks, floats and snprintf

4

Buffers in practice

UART output, a line-receive buffer and a command parser

Embedded C - ATmega328P - MPLAB X 2

2

A string is just a char array

```
char msg[6] = "Hello"; // 5 visible chars + 1 terminator
```

H

e

l

l

o

\0

[0]

[1]

[2]

[3]

[4]

[5]

The '\0' (value 0) marks the end. Every str* function walks memory until it finds one.

sizeof(msg) = 6
 Bytes reserved for the array, known at compile time.

strlen(msg) = 5
 Characters before the terminator, counted at run time.

Literals cost SRAM
 AVR is Harvard: "Hello" is copied from flash into SRAM at startup.

Embedded C - ATmega328P - MPLAB X 3

3

Why buffers matter on this chip

32 KB
Flash
Program code and constant data

2 KB
SRAM
Globals, literals, heap and stack

1 KB
EEPROM
Non-volatile settings

A 256-byte buffer is 12.5% of all RAM.

The stack grows down toward your globals. When they meet there is no exception and no warning: variables just get corrupted and the board resets or behaves strangely.

Embedded C - ATmega328P - MPLAB X

4

Declaring and sizing buffers

```
#define LINE_LEN 32 // longest msg + 1
char line[LINE_LEN]; // global: .bss,
static char tx[64]; // zeroed at reset
// file-private

void task(void) {
    char tmp[16]; // stack: garbage!
    tmp[0] = '\0'; // now empty string
    send(tmp, sizeof tmp);
}
```

Size = longest text + 1
Always budget the terminator.

Use sizeof(buf), not 16
One #define or sizeof keeps limits in sync.

Arrays decay to pointers
Inside a function, sizeof(ptr) is 2. Pass the length.

Embedded C - ATmega328P - MPLAB X

5

The <string.h> toolkit

FUNCTION	WHAT IT DOES	WATCH OUT
strlen(s)	Length, not counting '\0'	Undefined if s is unterminated
strcpy(d, s)	Copy s into d	No size check at all
strncpy(d, s, n)	Copy at most n chars	May leave d unterminated
strcat / strncat	Append s to end of d	n = chars added, not size of d
strcmp / strncmp	Compare; 0 means equal	Returns 0 on match, not 1
strchr / strstr	Find a char / substring	Returns NULL if not found
memset / memcpy	Fill / copy raw bytes	Ignores '\0'; you set the count

6

Copying safely: the strncpy trap

Risky

```
Char name[8];
strcpy(name, "Temperature");
// 12 bytes into 8: overflow

strcpy(name, src, sizeof name);
// fits, but no '\0' if
// src is 8+ chars long
```

Safe

```
strncpy(name, src, sizeof name - 1);
name[sizeof name - 1] = '\0';

// or, one line and always
// terminated:
snprintf(name, sizeof name,
         "%s", src);
```

Rule: any copy into a fixed buffer must know the buffer's size and guarantee a terminator.

Embedded C - ATmega328P - MPLAB X

7

7

Comparing and searching

```
if (strcmp(cmd, "STATUS") == 0) {
    // exact match
}
if (strncmp(cmd, "SET ", 4) == 0) {
    // prefix match
}
char *eq = strchr(line, '=');
if (eq) {
    // NULL check!
    *eq = '\0';
    // split in place
    key = line;    val = eq + 1;
}
```

== compares addresses

if (cmd == "STATUS") is almost never true.

strcmp returns 0 on match

if (strcmp(a, b)) means they differ.

Split by writing '\0'

Turns one buffer into two strings, zero copies.

key was = val

Embedded C - ATmega328P - MPLAB X

8

8

sprintf: formatting into a buffer

Turning numbers into text you can send, display or log

9

9

Anatomy of a sprintf call

```
int n = sprintf(buf, "T=%d C ID=%s", temp, id);
```

buf

Destination
char array you own. Must be big enough for the result + '\0'.

"T=%d..."

Format string
Literal text plus % specifiers, one per argument.

temp, id

Arguments
Matched left-to-right. Types must match specifiers.

n

Return value
Chars written, not counting the terminator.

T = 23 C ID = A7 \0 temp = 23, id = "A7" → n = 12

Embedded C - ATmega328P - MPLAB X

10

10

Format specifiers on an 8-bit AVR

SPEC	ARGUMENT TYPE	OUTPUT
%d %i	int (16-bit signed)	-1234
%u	unsigned int (16-bit)	65535
%ld %lu	long (32-bit)	100000
%x %02X	hex, zero-padded	3F
%c %s	char / string	A OK
%5d %-5d	width, alignment	uuu42

int is 16 bits here

```
uint32_t ms = 70000;
%d → garbage
%lu → 70000
```

Cast to be safe: (unsigned long)ms.
A mismatched specifier misreads the argument list and shifts every argument after it.

Embedded C - ATmega328P - MPLAB X

11

The floating-point gotcha

```
sprintf(buf, "%.2f", 3.14159);    // default AVR libc prints: ?
```

1 Link the float printf

Add to linker options:
-Wl,-u,vfprintf
-lprintf_float -lm
Works, but costs extra flash.

2 Use dtostrf()

From <stdlib.h>:
dtostrf(v, 6, 2, num);
then print num with %s.

3 Integer fixed-point

Store centi-degrees:
t = 2315 → "%d.%02d",
t / 100, t % 100
Handle sign separately.

Embedded C - ATmega328P - MPLAB X

12

12

snprintf: the bounded version

sprintf into char buf[8] with "ADC=10234"

A	D	C	=	1	0	2	3	4	\0
0	1	2	3	4	5	6	7	?	?

Bytes 8–9 land on whatever sits next to buf: another variable or a saved return address.

snprintf(buf, sizeof buf, ...)

A	D	C	=	1	0	2	\0
0	1	2	3	4	5	6	7

```
char buf[16];
int n = snprintf(buf,
    sizeof buf,
    "ADC=%u", val);

if (n >= (int)sizeof buf) {
    // truncated: n is the
    // length it WANTED
}

// writes at most size-1
// chars, always adds '\0'
```

Embedded C - ATmega328P - MPLAB X

13

13

Keep constant strings in flash

Flash

"ADC=%u\n" stored here



Startup copy

C runtime copies initialised data



SRAM

Same 9 bytes, now in RAM too

```
#include <avr/pgmspace.h>

// format string read from flash
sprintf_P(buf, PSTR("ADC=%u\n"), v);

// constant message table in flash
const char msg_ok[] PROGMEM = "OK";
strcpy_P(buf, msg_ok);
```

The _P family

sprintf_P, snprintf_P, strcpy_P, strcmp_P, strlen_P take a pointer to flash. Mixing them up reads the wrong memory space.

Embedded C - ATmega328P - MPLAB X

14

14

Putting it together: UART output

```
void uart_init(void) {
    UBRR0 = 103; // 9600 @ 16 MHz
    UCSRB = (1<<TXEN0)|(1<<RXEN0);
    UCSRC = (1<<UCSZ01)|(1<<UCSZ00);
}

// 8N1
void uart_putc(char c) {
    while (!(UCSR0A & (1<<UDRE0)));
    UDR0 = c;
}

void uart_puts(const char *s) {
    while (*s) uart_putc(*s++);
}
```

```
char buf[32];
uint16_t raw = adc_read(0);
uint32_t mv = raw * 5000UL / 1023;

snprintf(buf, sizeof buf,
    "A0: %1u.%031u V\n",
    mv / 1000, mv % 1000);
uart_puts(buf);
```

A0: 3.217 V

Embedded C - ATmega328P - MPLAB X

15

15

Receiving: a line buffer

```
#define RX_LEN 32
static char rx[RX_LEN];
static uint8_t rx_idx = 0;

void uart_poll(void) {
    if (!(UCSR0A & (1<<RXC0))) return;
    char c = UDR0;
    if (c == '\r' || c == '\n') {
        rx[rx_idx] = '\0';
        if (rx_idx) handle_cmd(rx);
        rx_idx = 0;
    } else if (rx_idx < RX_LEN - 1) {
        rx[rx_idx++] = c;
    } // else drop char
}
```

1 Byte arrives

RXC0 flag set in UCSROA

2 End of line?

Terminate with '\0', parse, reset index

3 Otherwise store

Only if a slot is left for '\0'

Embedded C - ATmega328P - MPLAB X

16

16

Parsing the command

```
void handle_cmd(char *cmd) {
    char out[24];
    if (strcmp(cmd, "READ") == 0) {
        snprintf(out, sizeof out,
            "ADC=%u\n", adc_read(0));
        uart_puts(out);
    } else if (strcmp(cmd, "LED ", 4) == 0) {
        if (atoi(cmd + 4) > 0) PORTB |= (1<<PB5);
        else PORTB &= ~(1<<PB5);
        uart_puts("OK\r\n");
    } else {
        uart_puts("ERR\r\n");
    }
}
```

Session

```
> READ
ADC=512
> LED 1
OK
> HELLO
ERR
```

Upgrade path: strtoll() reports where parsing stopped, so "LED x" can be rejected instead of read as 0.

Embedded C - ATmega328P - MPLAB X

17

17

Bugs to hunt in your own code

1 Off-by-one

char s[5] = "Hello"; leaves no room for '\0'.

2 Unterminated string

strncpy or a raw receive loop that never writes '\0'.

3 Wrong specifier

%d with a 32-bit value; use %lu or %ld.

4 %f prints ?

Float printf not linked. Use dtostrf or fixed-point.

5 Big local buffers

char tmp[200] in a function quietly eats the stack.

6 Comparing with ==

Compares addresses. Use strcmp(...) == 0.

Embedded C - ATmega328P - MPLAB X

18

18
