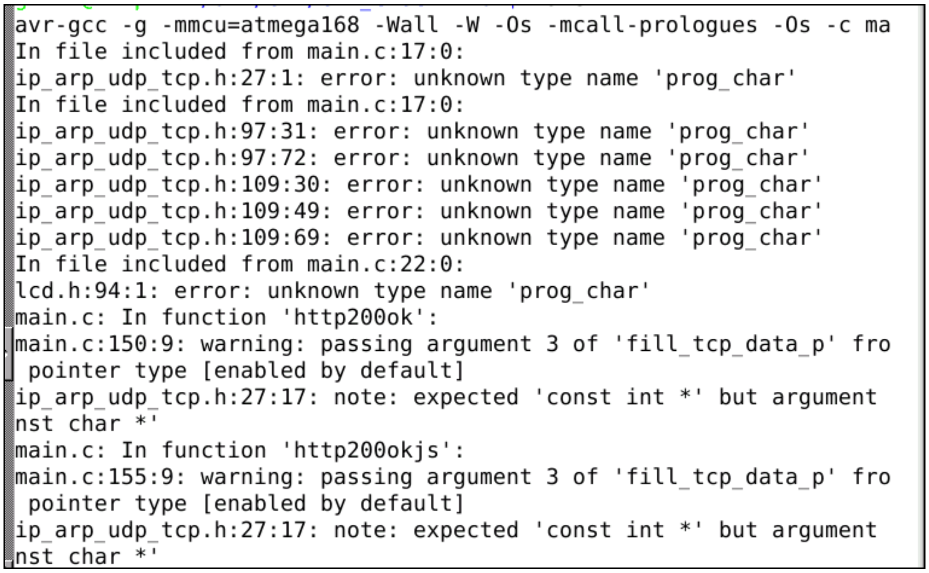
avr-gcc, error: unknown type name 'prog_char'



```
avr-gcc -g -mmcu=atmega168 -Wall -W -Os -mcall-prologues -Os -c ma
In file included from main.c:17:0:
ip_arp_udp_tcp.h:27:1: error: unknown type name 'prog_char'
In file included from main.c:17:0:
ip_arp_udp_tcp.h:97:31: error: unknown type name 'prog_char'
ip_arp_udp_tcp.h:97:72: error: unknown type name 'prog_char'
ip_arp_udp_tcp.h:109:30: error: unknown type name 'prog_char'
ip_arp_udp_tcp.h:109:49: error: unknown type name 'prog_char'
ip_arp_udp_tcp.h:109:69: error: unknown type name 'prog_char'
In file included from main.c:22:0:
lcd.h:94:1: error: unknown type name 'prog_char'
main.c: In function 'http200ok':
main.c:150:9: warning: passing argument 3 of 'fill_tcp_data_p' fro
 pointer type [enabled by default]
ip_arp_udp_tcp.h:27:17: note: expected 'const int *' but argument
nst char *'
main.c: In function 'http200okjs':
main.c:155:9: warning: passing argument 3 of 'fill_tcp_data_p' fro
 pointer type [enabled by default]
ip_arp_udp_tcp.h:27:17: note: expected 'const int *' but argument
nst char *'
```

Abstract:

Avr-libc has removed in version 1.8 the datatype prog_char. That is: as of atmel avrtoolchain-3.2.4 you will not be able to compile a lot of older code. Almost all tuxgraphics code is affected by this and we will update the code over time but it will take several month to do that.

This article explains how to compile old code with the new avr-libc without updating the code. It explains as well how the access to data in program memory should now be used with the new avr-libc.

How to compile old tuxgraphics code ?

Obviously one solution is to simply use the older compiler with the old code and it will work with no problem. You can find a link to the older avrtoolchain-3.2.3 further down.

The other solution is to add an option to the variable CFLAGS in the Makefile. This will enable the old behavior of the avr library and you will be able to **compile the old code with the new compiler**.
Look for a line starting with CFLAGS in the file Makefile and add "-Wno-deprecated-declarations -D__PROG_TYPES_COMPAT__" at the end.

Like this:

```
Change this line:
CFLAGS=-g -mmcu=$(MCU) -Wall -Wstrict-prototypes -Os -mcall-prologues

To:
CFLAGS=-g -mmcu=$(MCU) -Wall -Wstrict-prototypes -Os -mcall-prologues -Wno-deprecated-declarations -D__PROG_TYPES_COMPAT__
```

That's it! Just run "make" and you will be able to compile the code with no error or warning.

How to use the new "prog_char" when writing your own code?

The data type prog_char has disappeared! You **use now the compiler directive PROGMEM** to declare a variable that shell reside in program memory. Like this:

```
#include <avr/io.h>
#include <avr/pgmspace.h>
...
const char str_pstr[] PROGMEM = "FLASH STRING";
```

A function that takes such a datatype would the need to look like this:

```
void lcd_puts_p(const char *progstr )
{
        char c;
        while ((c = pgm_read_byte(progstr++))) {
                lcd_putc(c);
        }
}
```

It is a good convention to call such a function that takes a string from program memory "_p" in its name.

You would call the function from your code as follows:

```
lcd_puts_p(str_pstr);
```

There is as well the possibility to use a macro (the macro is part of the avr-libc) that will create a program memory string on the spot without the need to declare an own variable at the top of your code:

```
lcd_puts_p(PSTR("tuxgraphics"));
```

References/Download

- list of changes in avr-libc 1.8: http://www.nongnu.org/avr-libc/changes-1.8.html
- avrtoolchain-3.2.3, this is the older compiler and avr-libc which does still support the prog_char type: http://images.tuxgraphics.org/avrtoolchain-3.2.3/

2012-07-30, generated by tuxgrparser version 2.57