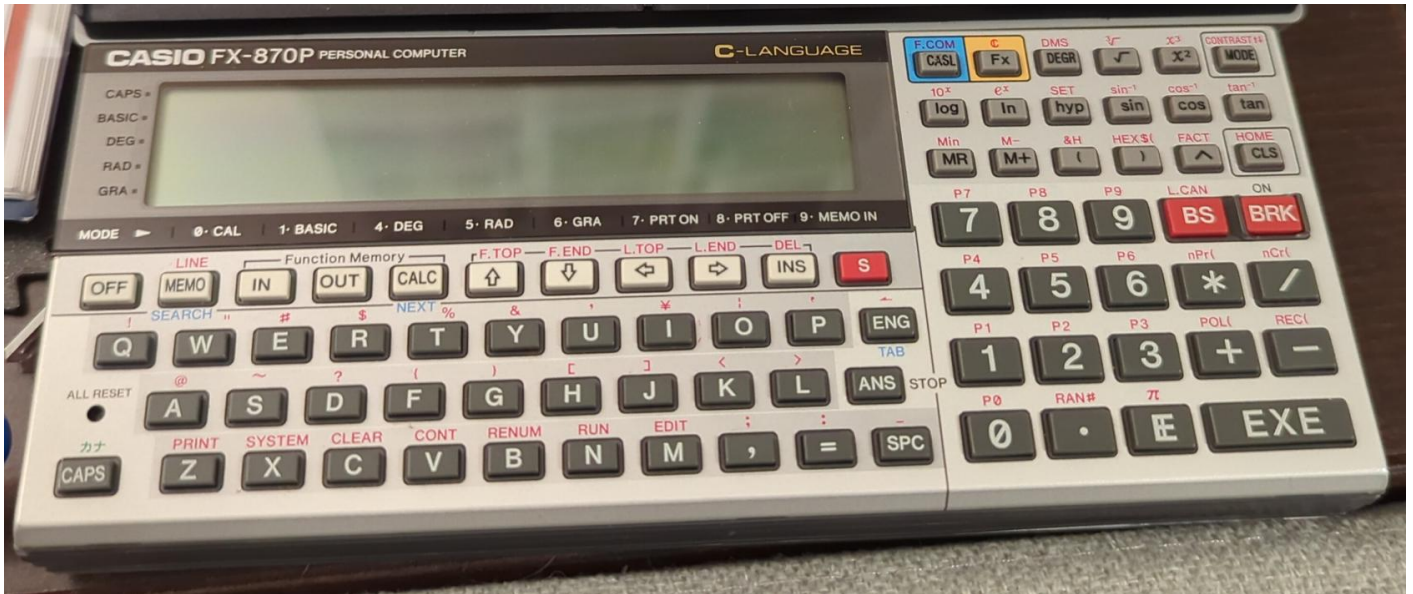


Casio FX-870P



Manual (Japanese with English commentary): [CasioVX-4-Manual-Peter-Rost.pdf](#)

Programming in C

Navigation

`S` (red), `F.COM` (blue) Enter C programming mode

`S` Source - edit code

`R` Run - compile and run code

Arrow keys Select program

(after program run) `S`, `M` or type `edit` Edit code

`S` (red), `MEMO` (line) Go to line

Types

`char` 8 bit

`short` 16 bit

`int` 16 bit

`unsigned` 16 bit

`long` 32 bit

`float` 32 bit

`double` 64 bit

`structs` or `unions` are not supported.

Flow control

```
if (C)
    expr;
```

```
if (C) {
    expr;
    expr;
}
```

```
if (C) {
    expr;
} else {
    expr;
}
```

```
while (C) {
    expr;
}
```

```
do {
    expr;
} while(C);
```

```
for (A; C; I) {  
    expr;  
}
```

```
goto LABEL;
```

```
LABEL;
```

`switch` / `case` are not supported.

Functions

Entry function:

```
main() {  
    expr;  
}
```

Custom function:

```
int foo(buf,x,y)  
    char buf[];  
    int x;  
    int y;  
{  
    int i;  
    expr;  
}
```

Variables need to be declared first then assigned.

Functions are declared in pre-ANSI style. Inner variables need to be defined at body top.

```
int getchar()
```

```
int getc(FILE) / int fgetc(FILE)
```

```
int putchar(char)
```

For `stdin` file: `extern FILE *stdin;`

<code>int putc(char, FILE) / int fputc(char, FILE)</code>	For <code>stdout</code> file: <code>extern FILE *stdout;</code> There is also <code>printer(?)</code> : <code>extern FILE *stdprn</code>
<code>char *gets(char *)</code> (unsafe)	<pre>char msg[8]; gets(msg); printf("Hello: %s",msg);</pre>
<code>char fgets(char *, int buflen, FILE)</code>	<pre>extern FILE *stdin; main() { char msg[8]; fgets(msg,8,stdin); printf("Hello: %s",msg); }</pre>
<code>int puts(char *)</code>	
<code>int fputs(char *,FILE)</code>	
<code>printf(format, args, ...)</code>	
<code>fprintf(FILE, format, args, ...)</code>	
<code>sprintf(char *buf, format, args, ...)</code>	
<code>int scanf(format, args, ...)</code>	
<code>int fscanf(FILE, format, args, ...)</code>	
<code>int sscanf(char *buf, format, args, ...)</code>	
<code>int fflush(FILE)</code>	
<code>int inport(int n)</code>	
<code>outport(int n, int i)</code>	
<code>clearerr(FILE in)</code>	
<code>breakpt()</code>	
<code>exit()</code> / <code>abort()</code>	

<code>char *malloc(unsigned size)</code>	
<code>char *calloc(unsigned n,unsigned size)</code>	
<code>int free(char *)</code>	
<code>int strlen(char *)</code>	
<code>char *strcpy(char *dest, char *src)</code>	
<code>char *?strcat(char *dest, char *src)</code>	
<code>int strcmp(char *, char *)</code>	
<code>char strchr(char *,char)</code>	
<code>int abs(int)</code>	
<code>double sin(double)</code> [<code>cos</code> , <code>tan</code> , <code>asin</code> , <code>acos</code> , <code>atan</code> , <code>sinh</code> , <code>cosh</code> , <code>tanh</code> , <code>asinh</code> , <code>acosh</code> , <code>atanh</code>]	
<code>double pow(double, double)</code>	
<code>double sqrt(double)</code>	
<code>double exp(double)</code>	
<code>double log(double)</code> / <code>double log10(double)</code>	
<code>angle(unsigned)</code>	mode for <code>sin</code> etc.: <code>0</code> - deg, <code>1</code> - rad, <code>2</code> - gra
<code>beep(unsigned)</code>	
<code>clrscr()</code>	
<code>gotoxy(unsigned x, unsigned y)</code>	<code>x</code> - between 0 and 31 (inc), <code>y</code> - between 0 and 3 (inc)