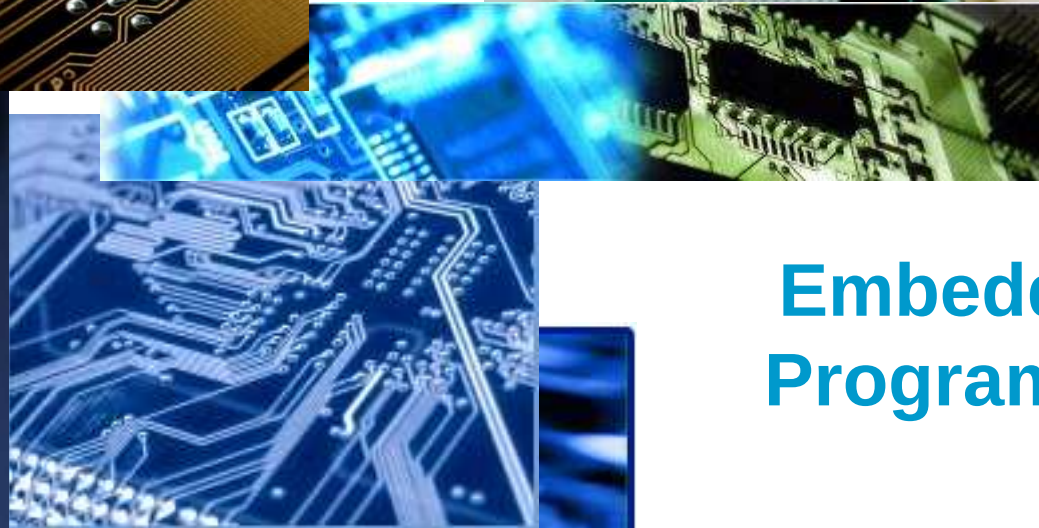
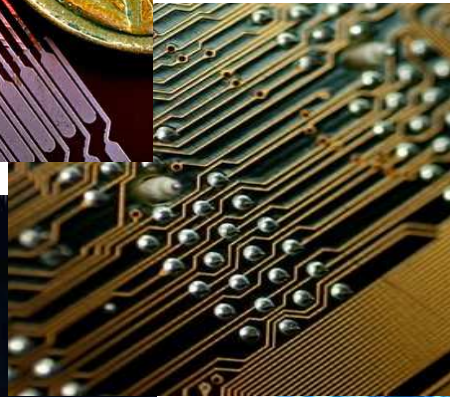
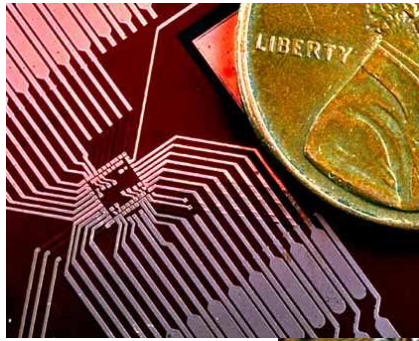




Embedded C Programming

Ramy Souody





Agenda:

- Introduction
- Compiler versus linker
- Operators
- Variables and Data Types
- Conditional Statements
- Loops
- Arrays
- Structures
- Interactive Questions



Agenda:



Introduction

- Compiler versus linker
- Operators
- Variables and Data Types
- Conditional Statements
- Loops
- Arrays
- Structures
- Interactive Questions

Introduction:

- What does the term Embedded C mean?
 - Embedded C is a proper subset from the C language suitable for Embedded Systems.
 - Some features exist in C and not used in Embedded C.

Introduction:

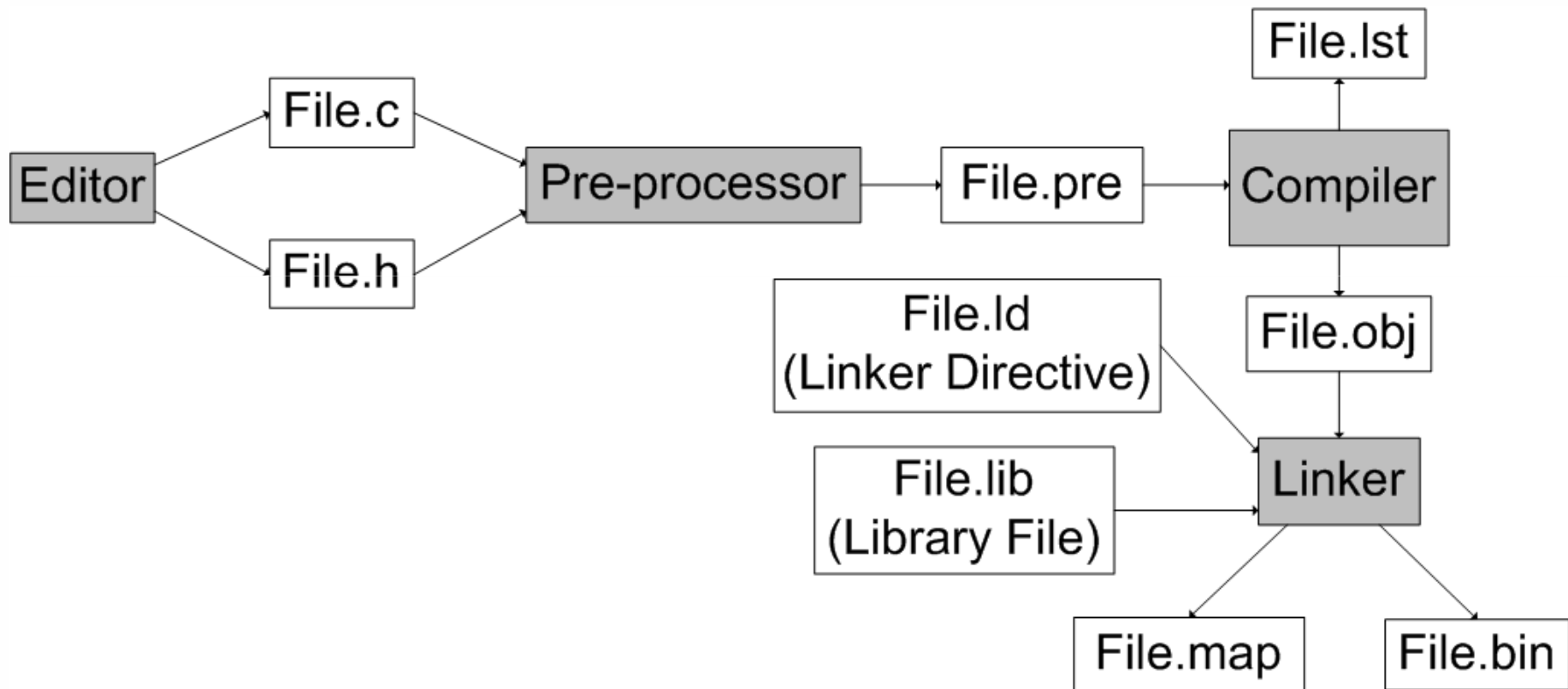
- Why do we use Embedded C?
 - Because Embedded Systems are not like general purpose computers since we have limited hardware resources.



Agenda:

- Introduction
-  Compiler versus linker
- Operators
- Variables and Data Types
- Conditional Statements
- Loops
- Arrays
- Structures
- Interactive Questions

Compiler versus linker:





Agenda:

- Introduction
- Compiler versus linker
-  Operators
 - Variables and Data Types
 - Conditional Statements
 - Loops
 - Arrays
 - Structures
 - Interactive Questions

Operators:

- Arithmetic Operators:
 - Binary Operators:
 - $x+y$ (Addition)
 - $x-y$ (Subtraction)
 - $x*y$ (Multiplication)
 - x/y (Division)
 - $x\%y$ (Reminder)

Operators:

- Arithmetic Operators:

- Unary Operators:

- $x++$ (Increment)
 - $x--$ (Decrement)

- Take Care :

- $x++$ (Postfix) {if $x=5$: $y=x++$; $y=5, x=6$ }
 - $++x$ (Prefix) {if $x=5$: $y=++x$; $y=6, x=6$ }

Operators:

- Assignment and Compound Operators:

– $x = y;$

– $x += y;$

– $x -= y;$

– $x *= y;$

– $x /= y;$

– $x \% = y;$

$x = x + y;$

$x = x - y;$

$x = x * y;$

$x = x / y;$

$x = x \% y;$

Operators:

- Relational Operators: (True or False)
 - $<$, $<=$, $>$, $>=$, $==$, $!=$
- Logical Operators: (True or False)
 - $\&\&$, $||$, $!$
- Bitwise Operators:
 - $\&$, $|$, $\wedge$ (XOR), $<<$, $>>$, $\sim$

Operators:

- Ternary Operator:
 - (condition) ? Statement_When_True : Statement_When_False
 - $(x < 10) ? (y = 5) : (y = 3);$
 - $y = (x < 10) ? 5 : 3;$

Operators:

- Precedence of Operations:
 - Casting
 - Parentheses ()
 - Negative
 - Multiplication and division
 - Addition and subtraction



Agenda:

- Introduction
- Compiler versus linker
- Operators
-  Variables and Data Types
 - Conditional Statements
 - Loops
 - Arrays
 - Structures
 - Interactive Questions

Variables and Data Types:

- Variable Modifier:
 - signed
 - unsigned
- Data Type:
 - char (8 bits)
 - short (16 bit)
 - long (32 bit)

Variables and Data Types:

- Variable Definition:
 - [Modifier] datatype variable_name [=initialization];
 - signed char x = 10;
- For each variable we should know:
 - Sign (signed or unsigned)
 - Size (char, short or long)
 - Variable name

Variables and Data Types:

- Naming rules for variables and functions:
 - Letters (A-Z and a-z)
 - Numbers (0-9)
 - Underscore (_)
 - Variable name must start with letter or underscore.

Variables and Data Types:

- Important tips:
 - No float:
 - Because the hardware must support floating point calculations (rarely to find in embedded systems).
 - No integer:
 - Because there is no rule for integer size, some compilers define integer numbers as 16 bits and others define them 32 bits.



Agenda:

- Introduction
- Compiler versus linker
- Operators
- Variables and Data Types
-  **Conditional Statements**
- Loops
- Arrays
- Structures
- Interactive Questions

Conditional Statements:

- If-else:

```
if(condition)
```

```
{.....;}
```

```
else
```

```
{.....;}
```

Conditional Statements:

- Nested if:

```
if(condition)
{.....;}
else {
    if(condition)
    {.....;}
    else
    {.....;}
}
```

Conditional Statements:

- Switch-case:

```
switch(x)
{
    case 0: {.....;}
    case 1: {.....;}
    [default: {.....;}]
}
```

Conditional Statements:

- Example: (Resort)
 - 100 LE Enter
 - 200 LE Enter + Club
 - 300 LE Enter + Club + Swimming Pool
 - 400 LE Enter + Club + Swimming Pool + Parking

Conditional Statements:

- Solution: (Resort)
switch(fees)
{
 case 400: parking;
 case 300: sw_pool;
 case 200: club;
 default: enter;
}

Agenda:

- Introduction
- Compiler versus linker
- Operators
- Variables and Data Types
- Conditional Statements



Loops

- Arrays
- Structures
- Interactive Questions

Loops:

- while: (Pre-condition)
while(condition)
{
 ;
 ;
}

Loops:

- do-while: (Post-condition)
do
{
.....;
.....;
} while(condition);

Loops:

- For:

```
for(initialization ; condition ; update)
```

```
{
```

```
.....;
```

```
.....;
```

```
}
```

```
– for(x=0,y=0 ; x<10 ; x--,y++)
```

```
{.....;}
```

Agenda:

- Introduction
- Compiler versus linker
- Operators
- Variables and Data Types
- Conditional Statements
- Loops
-  **Arrays**
- Structures
- Interactive Questions

Arrays:

- Array is a grouping of elements of the same type.
- Definition:
`char Arr[10]; //Array of 10 char elements`
- Initialization:
`char Arr[3]= {10,20,30};`
`char Arr[]={10,20,30};`

Arrays:

- Note:

The first element index in the array is 0 not 1.

- Example:

```
char Arr[3]= {10,20,30};
```

first element → Arr[0]=10

second element → Arr[1]=20

third element → Arr[2]=30

Arrays:

- Multi-Dimensional Array:

- 2D Array :

- short Arr[row][col];

- short Arr[2][2]={{1,2},{3,4}};

- 3D Array:

- short Arr[x][y][z];

- short Arr[2][2][2]={{1,2},{3,4}}, {{5,6},{7,8}}};



Agenda:

- Introduction
- Compiler versus linker
- Operators
- Variables and Data Types
- Conditional Statements
- Loops
- Arrays
-  Structures
- Interactive Questions

Structures:

- Structure is a complex data type to group elements of different types.
- Declaration:

```
struct Str_Name  
{  
    datatype Var_Name;  
};
```

Structures:

- Example:

```
struct student
```

```
{
```

```
    char name[20];
```

```
    short id;
```

```
};
```

(This is only a declaration
and doesn't reserve any
memory space)

Structures:

- Definition:

```
main()
{
    short id;
    struct student S1; // here we reserved memory
                        space for the structure S1 of type student
    S1.id=20;
    id=30;
}
```

Structures:

- Notes:
 - It is legal to name variables in different structures by the same name.
 - Structure can include another structure but must be declared first.
 - Structure declaration doesn't reserve any memory space.

Agenda:

- Introduction
- Compiler versus linker
- Operators
- Variables and Data Types
- Conditional Statements
- Loops
- Arrays
- Structures
-  Interactive Questions

Interactive Questions

- what will be the output of the following program?

```
main()
{
    int i=4;
    switch(i)
    {
        default: printf("default");
        case 1: printf("case 1");
                break;
        case 2: printf("case 2");
                break;
    }
}
```

Interactive Questions

- point out the error, if any, in the for loop

```
main()
```

```
{
```

```
    int i=1;
```

```
    for ( ; ; )
```

```
    {
```

```
        i++;
```

```
        if(i>10)break;
```

```
    }
```

```
}
```

- a. the condition is a must
- b. the two semicolons should be dropped
- c. replace for loop with while loop
- d. no error

Interactive Questions

- point out the error, if any, in the while loop

```
main()
```

```
{
```

```
    int i=1;
```

```
    while( )
```

```
    {
```

```
        i++;
```

```
        if(i>10)break;
```

```
    }
```

```
}
```

- a. the condition is a must
- b. there should be a semicolon in while()
- c. replace while loop with for loop
- d. no error

Interactive Questions

- point out the error, if any, in the following program

```
main()
{
    int i=4,j=2;
    switch(i)
    {
        case 1: printf("case 1");
                break;
        case j: printf("case 2");
                break;
    }
}
```

Interactive Questions

- point out the error, if any, in the following program

```
main()
{
    int i=1;
    switch(i)
    {
        case 1: printf("case 1");
                break;
        case 1*2+4: printf("case 2");
                break;
    }
}
```

Interactive Questions

- point out the error, if any, in the following program

```
main()
{
    int a=10,b;
    a>=5? b=100: b=200;
}
```

Interactive Questions

- what will be the output of the following program?

```
main()
{
    int Arr[2][2][2]={10,20,30,40,50,60,70,80};
    printf("%d",Arr[1][1][1]);
}
```



info@escommittee.com

education@escommittee.com