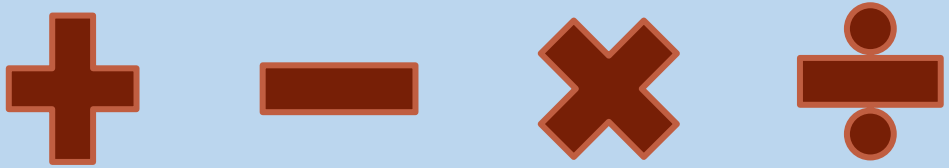
Wels

C Embebido

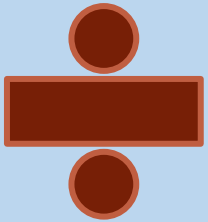
Operadores matemáticos



¡Veamos un ejemplo!

Wels

División



c = 2.000000 ✗

Because: int / int ➡ int

c = 2.500000 ✓

Because: float / int ➡ float

Wels

Casteo en C

Se refiere a cambiar una variable de un tipo a otro.

El Casting (casteo) permite hacer explícita o forzar esta conversión.

Veremos dos tipos:

- Tipo implícita
- Tipo explícita

Casteo: Tipo Implícita

El compilador realiza automáticamente la conversión de tipo.

```
int  data = 17;
char caract = '0'; /* ASCII valor es: 48 */
int sum_t;

sum_t = data + caract;
printf("Value of sum : %d\n", sum_t );
printf("-----------------------------\r\n");
```

Casteo: Tipo Explícita

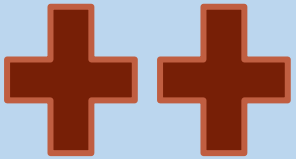
El programador plantea el tipo de conversión que se realiza.

(**tipo dato**) expresión

```
uint32_t sum = 17, count = 5;
float z;

z = (float)sum/count;

printf("Resultado de la operacion: %.3f\r\n",z);
```

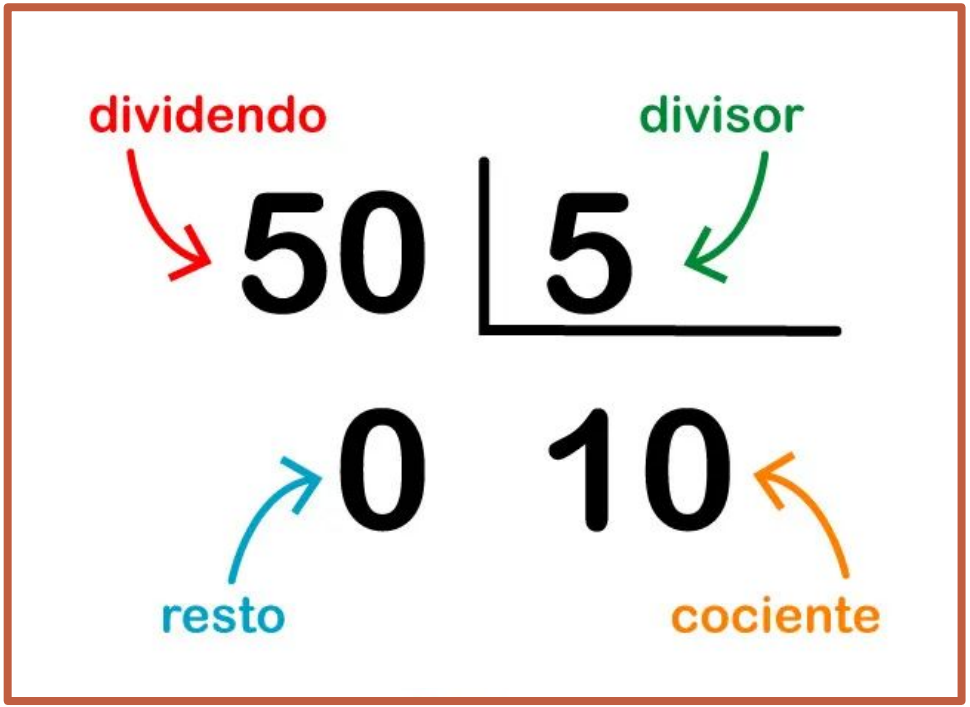
Wels

Operadores matemáticos

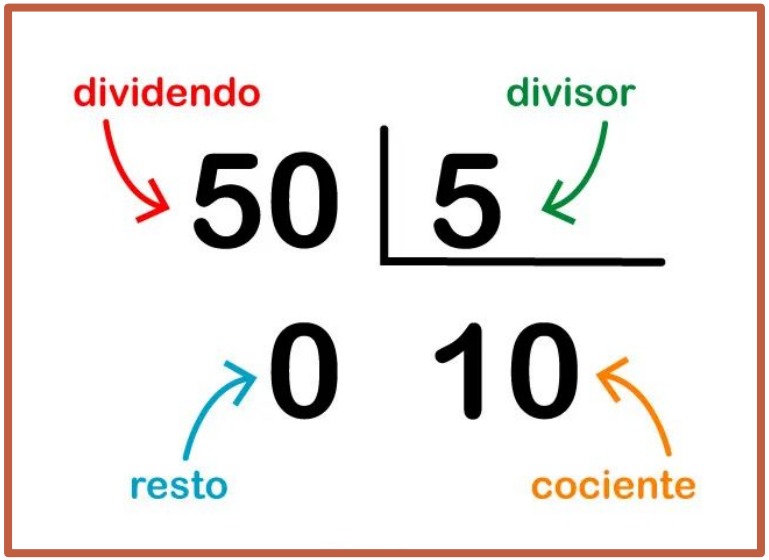


Módulo - Incremento -Decremento

Módulo %

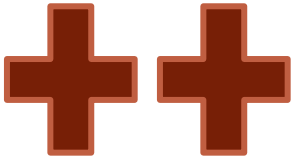


Módulo



- **El operador de módulo**, denotado por %, es un operador aritmético.
- El operador de módulo produce el resto de una división entera.
- El operador % no se puede aplicar a números float o double.

Incremento

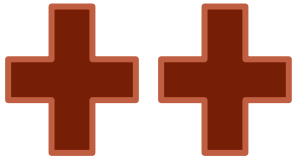


El operador de incremento, denotado por ++, se usa para incrementar el valor de una variable en 1.

Pre-Incremento

Post-Incremento

Incremento



Pre-Incremento

```
x = 5;
y = (++x) + 5;
```

Post-Incremento

```
x = 5;
y = (x++) + 5;
```

Decremento

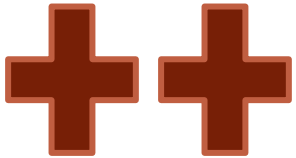
Wels

El operador de decremento, denotado por --, se usa para decrementar el valor de una variable en 1.

Pre-Decremento

Post-Decremento

Decremento

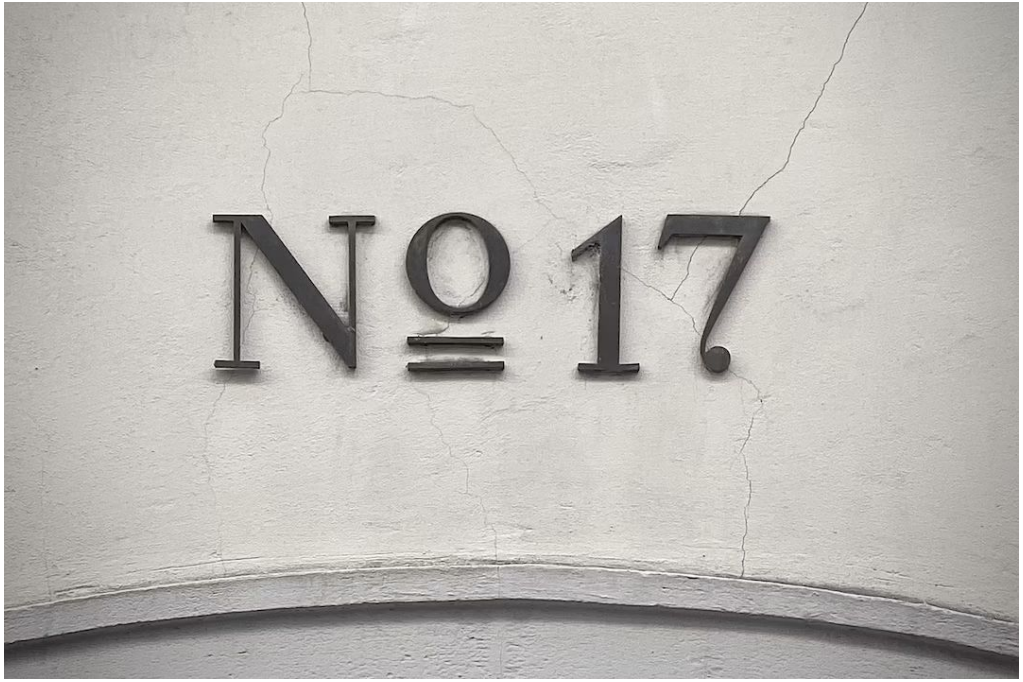


Pre-Decremento

```
x = 5;
y = (--x) + 5;
```

Post-Decremento

```
x = 5;
y = (x--) + 5;
```



Dirección

Punteros

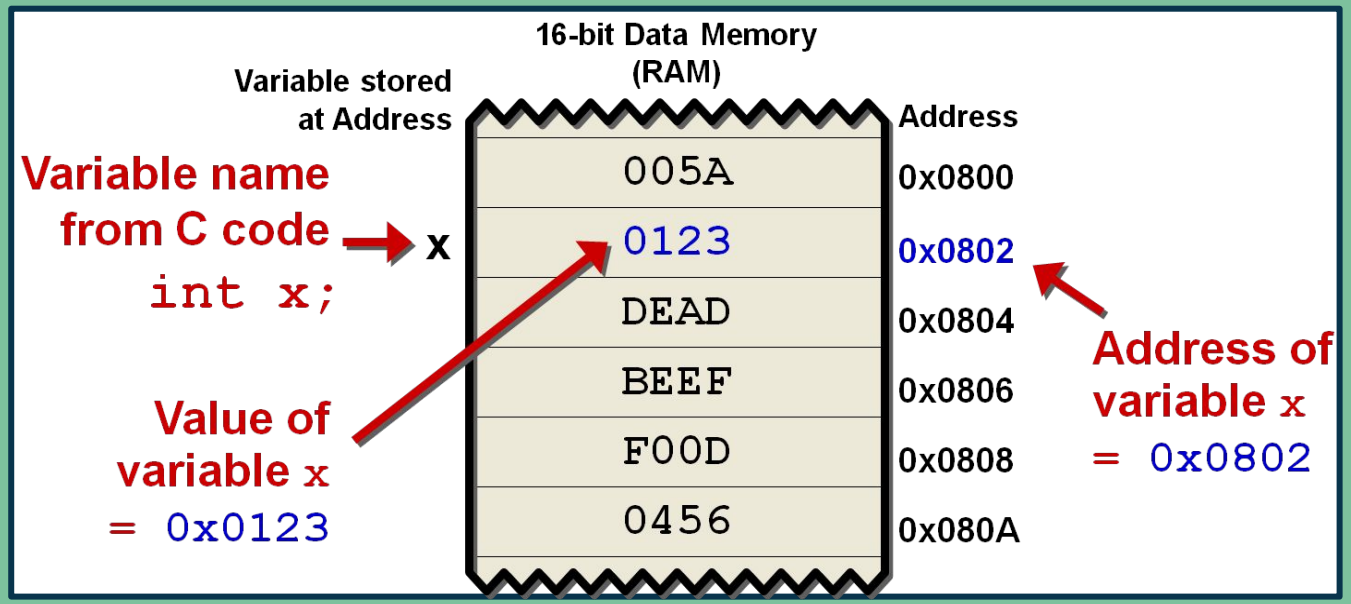
Wels

Tipos de datos

Wels

Address

&Variable



Wels

Address

&Variable

- Cada variable declarada y definida se almacena en la memoria.
- Para conocer la dirección usamos “**&**” adelante de la variable.

```
uint32_t data = 15;
&data;
```

Wels

Pointer

*p

- Un puntero contiene la dirección de otra variable o función.
- Permite un acceso indirecto a la variable.

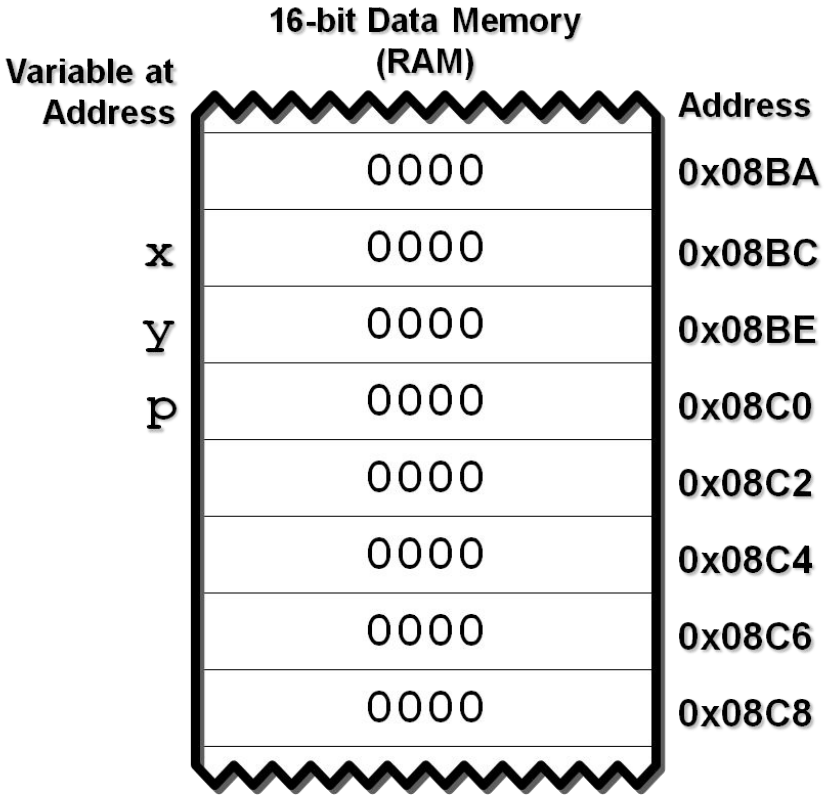
```
uint32_t *p;
```

El operador “*” indica que es un puntero a

Ejemplo de Punteros

Wels

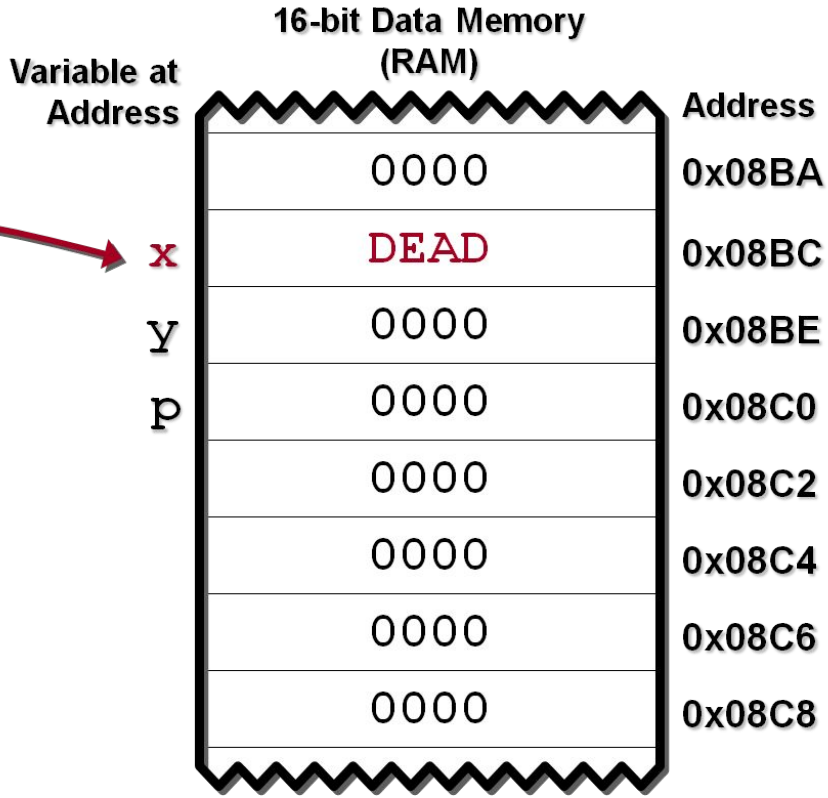
```
{
  int x, y;
  int *p;
```



Ejemplo de Punteros

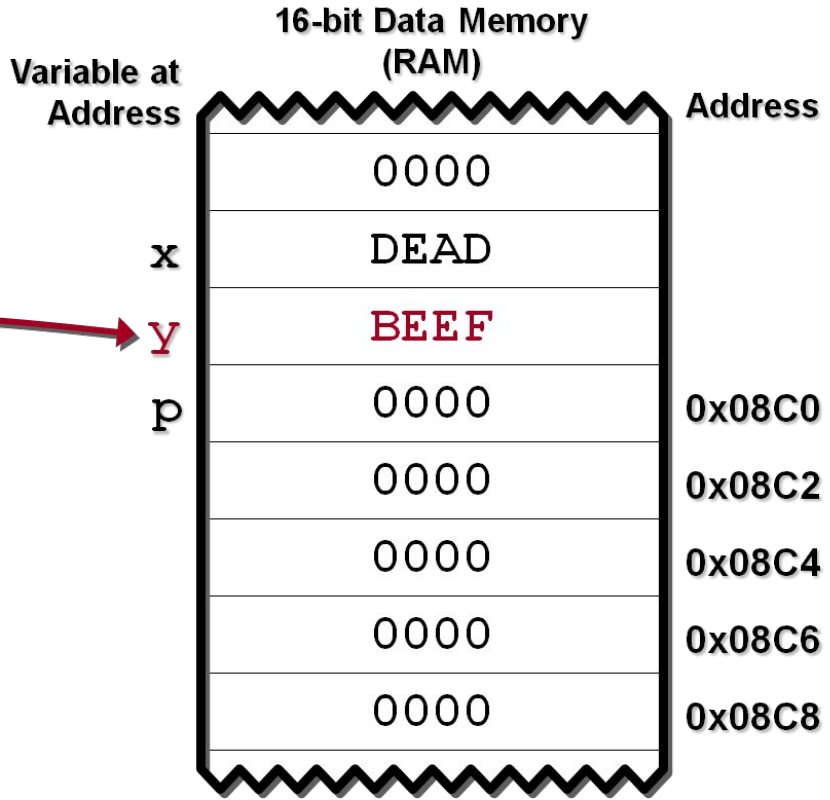
Wels

```
{
  int x, y;
  int *p;

  x = 0xDEAD;
```



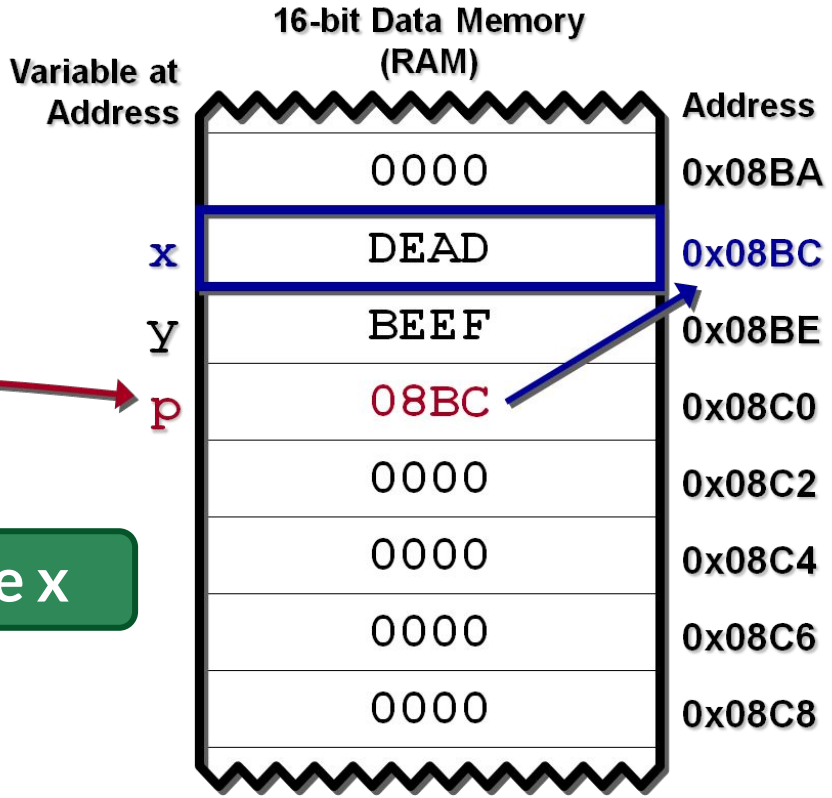
Ejemplo de Punteros

```
{
  int x, y;
  int *p;

  x = 0xDEAD;
  y = 0xBEEF;
```



Ejemplo de Punteros

```
{
  int x, y;
  int *p;

  x = 0xDEAD;
  y = 0xBEEF;
  p = &x;

}
```

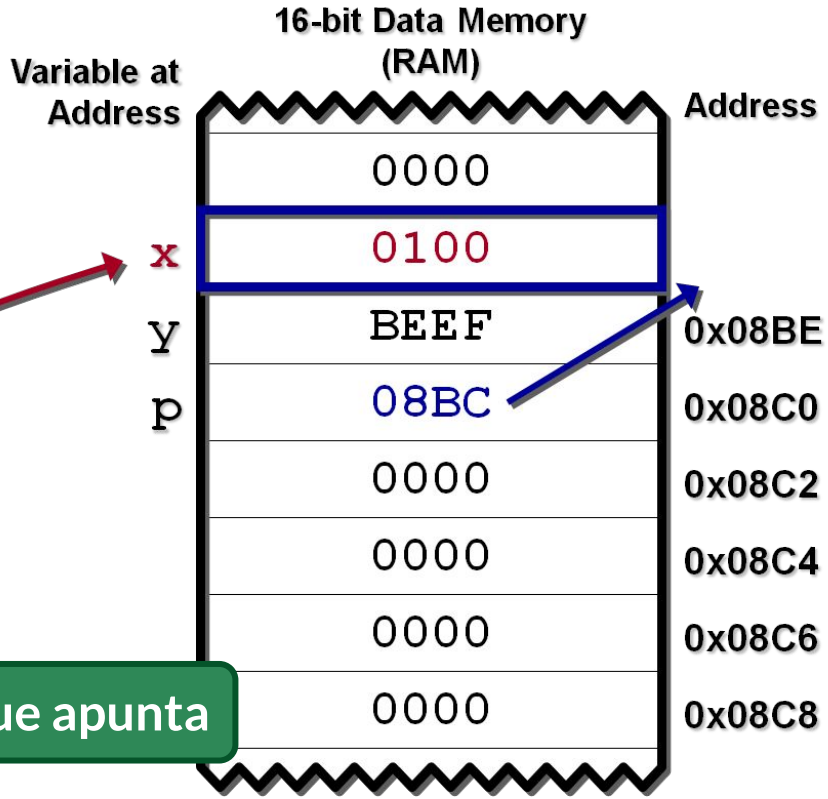


Le asignamos la dirección de x

Ejemplo de Punteros

Wels

```
{
  int x, y;
  int *p;

  x = 0xDEAD;
  y = 0xBEEF;
  p = &x;

  *p = 0x0100;
```

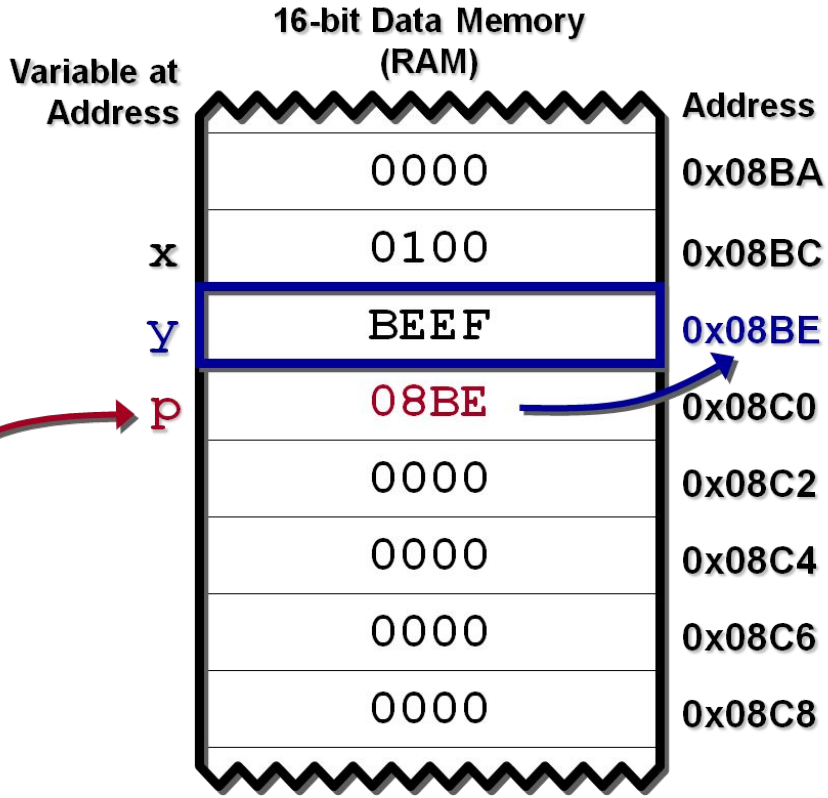


Asignar el valor a la variable que apunta

Ejemplo de Punteros

Wels

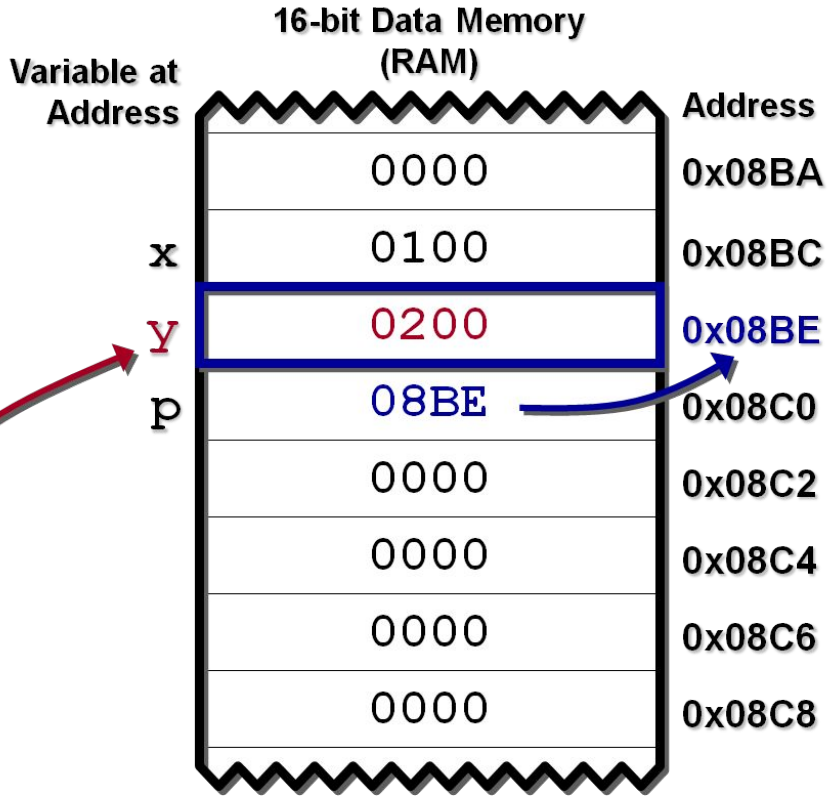
```
{
  int x, y;
  int *p;

  x = 0xDEAD;
  y = 0xBEEF;
  p = &x;

  *p = 0x0100;
   p = &y;
  *p = 0x0200;
}
```



Ejemplo de Punteros

Wels

```
{
  int x, y;
  int *p;

  x = 0xDEAD;
  y = 0xBEEF;
  p = &x;

  *p = 0x0100;
   p = &y;
  *p = 0x0200;
}
```



Wels

Pointer

*p

- Un puntero contiene la dirección de otra variable o función.
- Permite un acceso indirecto a la variable.

```
uint32_t *p;
```

El operador “*” indica que es un puntero a

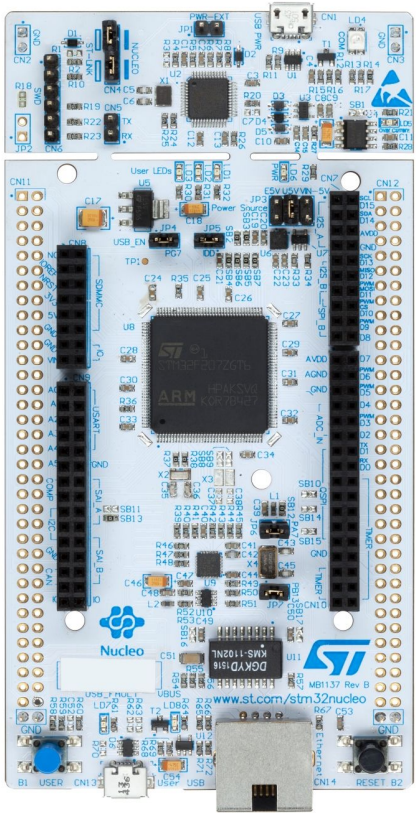
Wels

Pointer

¿Cómo se usa?

- Almacenar data en RAM.
- Copiar data de Registros a memoria RAM y viceversa.
- Configurar los registros de periféricos.
- Punteros a ISR, para manejo de interrupciones.

Wels



Gracias

@welstheory
hola@welstheory.com
+51 918 899 684