

PUNTEROS EN C

Ejemplos en Sistemas Embebidos
¿Qué es?

Definir un puntero - Operadores

Modificando valores indirectamente

RP2040

si queremos utilizar el puerto I2C, dentro de la documentación vemos que se inicializa asi:

```
uint i2c_init (i2c_inst_t *i2c, uint baudrate)
```

Donde i2c_inst_t es un tipo de dato y el operador "*" indica que se le debe pasar un puntero

FreeRtos:

Así como existe Windows o Android, dentro de los SE, se pueden utilizar sistemas operativos. Uno de los más famosos es FreeRtos, y para definir una tarea que deba realizar se le deben pasar un parámetro como puntero:

```
Void Task1_freeRtos ( void *parameters)
```

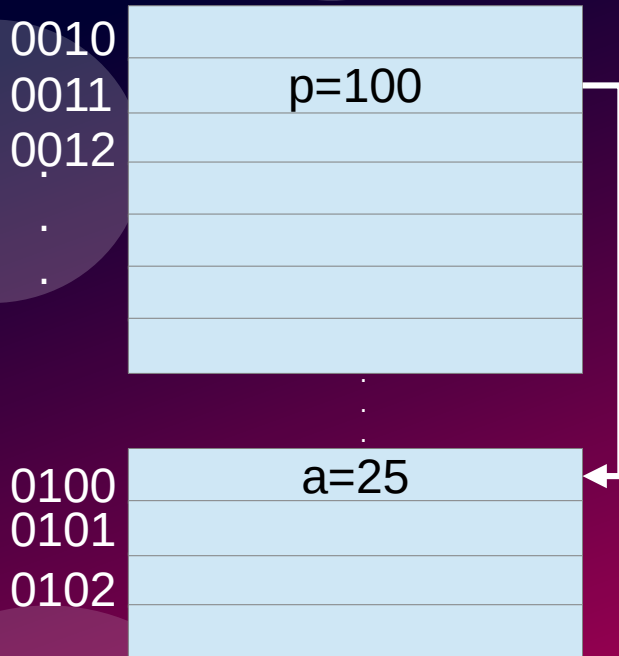
STM32

La familia de uC de la marca STM32, provee una biblioteca de código en C, llamada "HAL", esta en casi todas sus versiones se maneja mediante punteros, por ejemplo para inicializar un I2C de alguno de sus microcontroladores de 32 bits:

```
void I2C_DMAMasterTransmitCplt(DMA_HandleTypeDef *hdma);
```

¿Que es un puntero en C?

- Un puntero es una variable que almacena una posición de memoria en lugar de un dato.



Definir un puntero en C

```
tipo_dato * nombre_puntero
```

Por ejemplo:

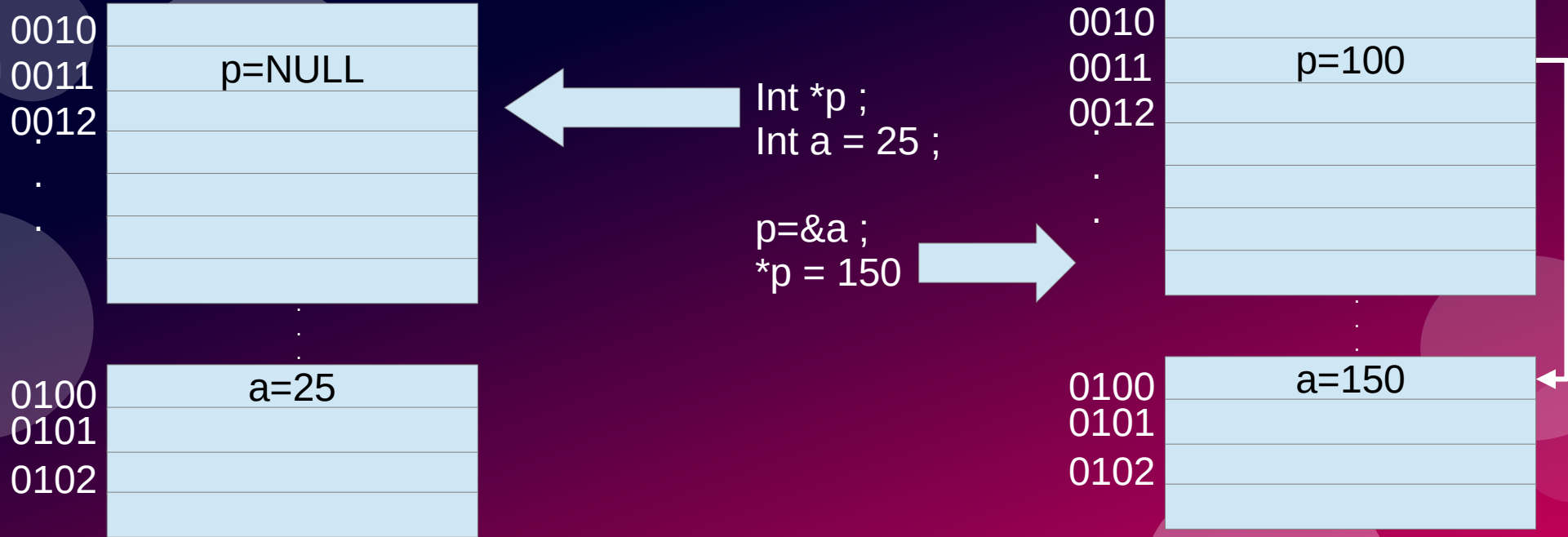
- `int *ptr1`
- `char *puntero2`
- `float *ptrf`

Operadores para punteros

- Operador & o dirección: Obtiene la posición de memoria de una variable: por ejemplo `printf("%ud",&a)` imprime la posición de memoria de a.
- Operador indirección "*" o puntero: imprime el valor de la posición de memoria de a.

Vamos a verlo con un ejemplo:

MODIFICANDO VALORES INDIRECTAMENTE



Vamos a verlo con un ejemplo:

MODIFICAR VALORES INDIRECTAMENTE

- Se tiene un valor `int d1 = 100`
- Creamos un puntero a entero:
`int *ptr`
- Apuntamos `ptr` a `d1`: `ptr=&d1`
- Modificamos `d1`: `*ptr = 180`
- Ahora `d1` vale 180 sin haber realizado explícitamente
“`d1=180`”

Tarea:

- Escriba un programa que realice esto