



RoboKit

ESTRUCTURAS DE CÓDIGO CON SENTIDO DEL COMPILADOR

Las estructuras de código con sentido son generadas a través de los tipos de datos, dónde se evalúan las estructuras ya generadas para que tengan un sentido que es aceptado por este compilador, por ende las estructuras con sentido de este compilador que se pueden generar son las que están en la siguiente tabla:

Estructuras de Declaraciones de puerto (DP) proximity	
PUERTO proximity IDENTIFICADOR FIN_DE_LINEA	port_A1 proximity proximitySensor!
Estructuras de Declaraciones de puerto (DP) temperature	
PURTO temperature IDENTIFICADOR FIN_DE_LINEA	port_A2 temperature temperatureSensor!
Estructuras de Declaraciones de puerto (DP) LED	
PUERTO LED IDENTIFICADOR FIN_DE_LINEA	port_A3 LED LEDSensor!
Estructuras de Declaraciones de puerto (DP) LCD	
PUERTO LCD IDENTIFICADOR FIN_DE_LINEA	port_A4 LCD LCDSensor!
Estructuras de Declaraciones de puerto (DP) button	
PUERTO button IDENTIFICADOR FIN_DE_LINEA	port_A5 button buttonSensor!
Estructuras de Declaraciones de puerto (DP) motor	
PUERTO motor IDENTIFICADOR FIN_DE_LINEA	port_B6 motor motorSensor!
Estructuras de Declaraciones con valor (DCV) string	
DECLARACION string IDENTIFICADOR OPERADOR_ASIGNACION IDENTIFICADOR_CADENA FIN_DE_LINEA	var string holaMundo = 'HolaMundo'!
Estructuras de Declaraciones con valor (DCV) int	

DECLARACION int IDENTIFICADOR OPERADOR_ASIGNACION NUMERO_ENTERO FIN_DE_LINEA	var int numero1 = 234!
Estructuras de Declaraciones con valor (DCV) decimal	
DECLARACION decimal IDENTIFICADOR OPERADOR_ASIGNACION NUMERO_REAL FIN_DE_LINEA	var decimal numero2 = 34.56!
Estructuras de Declaraciones con valor (DCV) boolean	
DECLARACION boolean IDENTIFICADOR OPERADOR_ASIGNACION CONDICIONAL FIN_DE_LINEA	var boolean valor3 = true!
Estructuras de Declaraciones sin valor (DSV) string	
DECLARACION string IDENTIFICADOR FIN_DE_LINEA	const string valor1 !
Estructuras de Declaraciones sin valor (DSV) int	
DECLARACION int IDENTIFICADOR FIN_DE_LINEA	const int valor2 !
Estructuras de Declaraciones sin valor (DSV) decimal	
DECLARACION decimal IDENTIFICADOR FIN_DE_LINEA	const decimal valor3 !
Estructuras de Declaraciones sin valor (DSV) boolean	
DECLARACION boolean IDENTIFICADOR FIN_DE_LINEA	const boolean valor4 !
Estructura de Asignaciones (A) string	
IDENTIFICADOR OPERADOR_ASIGNACION IDENTIFICADOR_CADENA FIN_DE_LINEA	Valor1 = 'Hola' !
Estructura de Asignaciones (A) int	
IDENTIFICADOR OPERADOR_ASIGNACION NUMERO_ENTERO FIN_DE_LINEA	Valor2 = 3 !
Estructura de Asignaciones (A) decimal	

IDENTIFICADOR OPERADOR_ASIGNACION NUMERO_REAL FIN_DE_LINEA	valor3 = 3.3 !
Estructura de Asignaciones (A) boolean	
IDENTIFICADOR OPERADOR_ASIGNACION CONDICIONAL FIN_DE_LINEA	Valor4 = true !
Estructura de funciones de los motores con valor (FMCV) move	
move SIGNO_AGRUPACION_3 IDENTIFICADOR SIGNO_PUNTUACION_1 NUMERO_ENTERO SIGNO_PUNTUACION_1 IDENTIFICADOR SIGNO_AGRUPACION_4 FIN_DE_LINEA	move(motorSensor,34,button1)!
Estructura de funciones de los motores sin valor (FMCV) start	
start SIGNO_AGRUPACION_3 IDENTIFICADOR SIGNO_PUNTUACION_1 IDENTIFICADOR SIGNO_AGRUPACION_4 FIN_DE_LINEA	start(motorSensor,button1)!
Estructura de funciones de los motores sin valor (FMCV) restart	
restart SIGNO_AGRUPACION_3 IDENTIFICADOR SIGNO_PUNTUACION_1 IDENTIFICADOR SIGNO_AGRUPACION_4 FIN_DE_LINEA	restart(motorSensor,button1)!
Estructura de Método de impresora a consola (MIC)	
print SIGNO_AGRUPACION_3 console SIGNO_PUNTUACION_1 (IDENTIFICADOR _CADENA IDENTIFICADOR) SIGNO_AGRUPACION_4 FIN_DE_LINEA	print(console,'El motor1 se esta moviendo')!
Estructura de Método de impresora a LCD (MIL)	
print SIGNO_AGRUPACION_3 IDENTIFICADOR SIGNO_PUNTUACION_1 (IDENTIFICADOR _CADENA IDENTIFICADOR) SIGNO_AGRUPACION_4 FIN_DE_LINEA	print(LCDSensor,'El motor1 se esta moviendo')!
Estructura de Operaciones (O)	
operation IDENTIFICADOR OPERADOR_ASIGNACION (NUMERO_ENTERO NUMERO_REAL 	operation valor2 = a + b!

IDENTIFICADOR) OPERADOR_ARITMETICO (NUMERO_ENTERO NUMERO_REAL IDENTIFICADOR) FIN_DE_LINEA	
Estructura de las funciones sin parámetros (FSP)	
function IDENTIFICADOR SIGNO_AGRUPACION_3 SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (FMCV FMSV MIC MIL)* SIGNO_AGRUPACION_2	function moverMotor(){ move(motorSensor,60)! print(console,'El motor1 se esta moviendo')! }
Estructura de las funciones con parámetros (FCP) int	
function int IDENTIFICADOR SIGNO_AGRUPACION_3 int IDENTIFICADOR (SIGNO_PUNTUACION_1 int IDENTIFICADOR)* SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (O)* return IDENTIFICADOR FIN_DE_LINEA SIGNO_AGRUPACION_2	function int sumaNumerosInt(int a, int b){ operation valor2 = a + b! return valor2! }
Estructura de las funciones con parámetros (FCP) decimal	
function decimal IDENTIFICADOR SIGNO_AGRUPACION_3 decimal IDENTIFICADOR (SIGNO_PUNTUACION_1 decimal IDENTIFICADOR)* SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (O)* return IDENTIFICADOR FIN_DE_LINEA SIGNO_AGRUPACION_2	function decimal sumaNumerosInt(decimal a, decimal b){ operation valor2 = a + b! return valor2! }
Estructura de llamadas de funciones con valor (LFCV)	
call IDENTIFICADOR SIGNO_AGRUPACION_3 (NUMERO_ENTERO NUMERO_REAL) (SIGNO_PUNTUACION_1 (NUMERO_ENTERO NUMERO_REAL))* SIGNO_AGRUPACION_4 FIN_DE_LINEA	call sumaNumerosInt (5,10)!
Estructura de llamadas de funciones sin valor (LFSV)	
call IDENTIFICADOR SIGNO_AGRUPACION_3 SIGNO_AGRUPACION_4 FIN_DE_LINEA	call moverMotor()!
Estructura de control Begin (B)	



RoboKit

begin SIGNO_AGRUPACION_1 (LFCV LFSV FMCV FMSV)* SIGNO_AGRUPACION_2	begin{ call moverMotor()! }
Estructura de Método de los sensores (MS)	
call IDENTIFICADOR OPERADOR_ASIGNACION IDENTIFICADOR SIGNO_PUTUACION_2 METODO_SENSOR SIGNO_AGRUPACION_3 SIGNO_AGRUPACION_4 FIN_DE_LINEA	call Distancia = proximitySensor.distance()!
Estructura de Método Delay (MD)	
delay SIGNO_AGRUPACION_3 (IDENTIFICADOR NUMERO_ENTERO) SIGNO_AGRUPACION_4 FIN_DE_LINEA	delay(1000)!
Estructura de Método Encender (ME)	
ligther SIGNO_AGRUPACION_3 IDENTIFICADOR SIGNO_PUNTUACION_1 IDENTIFICADOR SIGNO_AGRUPACION_4 FIN_DE_LINEA	ligther(led1,boton1)!
Estructura de Método Apagar (MA)	
turn_off SIGNO_AGRUPACION_3 IDENTIFICADOR SIGNO_PUNTUACION_1 IDENTIFICADOR SIGNO_AGRUPACION_4 FIN_DE_LINEA	turn_off(led1,boton1)!
Estructura de control If Relacional (IR)	
if SIGNO_AGRUPACION_3 (IDENTIFICADOR NUMERO_ENTERO NUMERO_REAL) OPERADOR_RELACIONAL (IDENTIFICADOR NUMERO_ENTERO NUMERO_REAL) SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2	if(numero1>valor2){ call moverMotor()! }
Estructura de control If Lógico con dos valores (IL2V)	
if SIGNO_AGRUPACION_3 (IDENTIFICADOR CONDICIONAL) OPERADOR_LOGICO (IDENTIFICADOR CONDICIONAL) SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (MS MIC MIL A	if(valor3 and valor4){ call moverMotor()! }

DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2	
Estructura de control If Lógico con un valor (IL1V)	
if SIGNO_AGRUPACION_3 (IDENTIFICADOR CONDICIONAL) SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2	<pre>if(valor3){ call moverMotor()! }</pre>
Estructura de control Else Relacional (ER)	
if SIGNO_AGRUPACION_3 (IDENTIFICADOR NUMERO_ENTERO NUMERO_REAL) OPERADOR_RELACIONAL (IDENTIFICADOR NUMERO_ENTERO NUMERO_REAL) SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2 else SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2	<pre>if(numero1>valor2){ call moverMotor()! }else{ }</pre>
Estructura de control Else Lógico con dos valores (EL2V)	
if SIGNO_AGRUPACION_3 (IDENTIFICADOR CONDICIONAL) OPERADOR_LOGICO (IDENTIFICADOR CONDICIONAL) SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2 else SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2	<pre>if(valor3 and valor4){ } }else{ call moverMotor()! }</pre>
Estructura de control Else Lógico con un valor (EL1V)	
if SIGNO_AGRUPACION_3 (IDENTIFICADOR CONDICIONAL) SIGNO_AGRUPACION_4 SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2 else SIGNO_AGRUPACION_1 (MS MIC MIL A DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2	<pre>if(valor3){ call moverMotor()! }else{ }</pre>
Estructura de control Loop (L)	

RoboKit



```
loop SIGNO_AGRUPACION_1 (MS | MIC |  
MIL | A | DCV | DSV | FMCV | FMSV | LFCV |  
LFSV | IR | IL2V | IL1V | ER | EL2V | EL1V |  
MD | ME | MA)* SIGNO_AGRUPACION_2
```

```
loop{  
  if(valor3 and valor4){  
    call moverMotor()!  
  }else{  
  
  }  
  call Distancia = proximitySensor.distance()!  
}
```