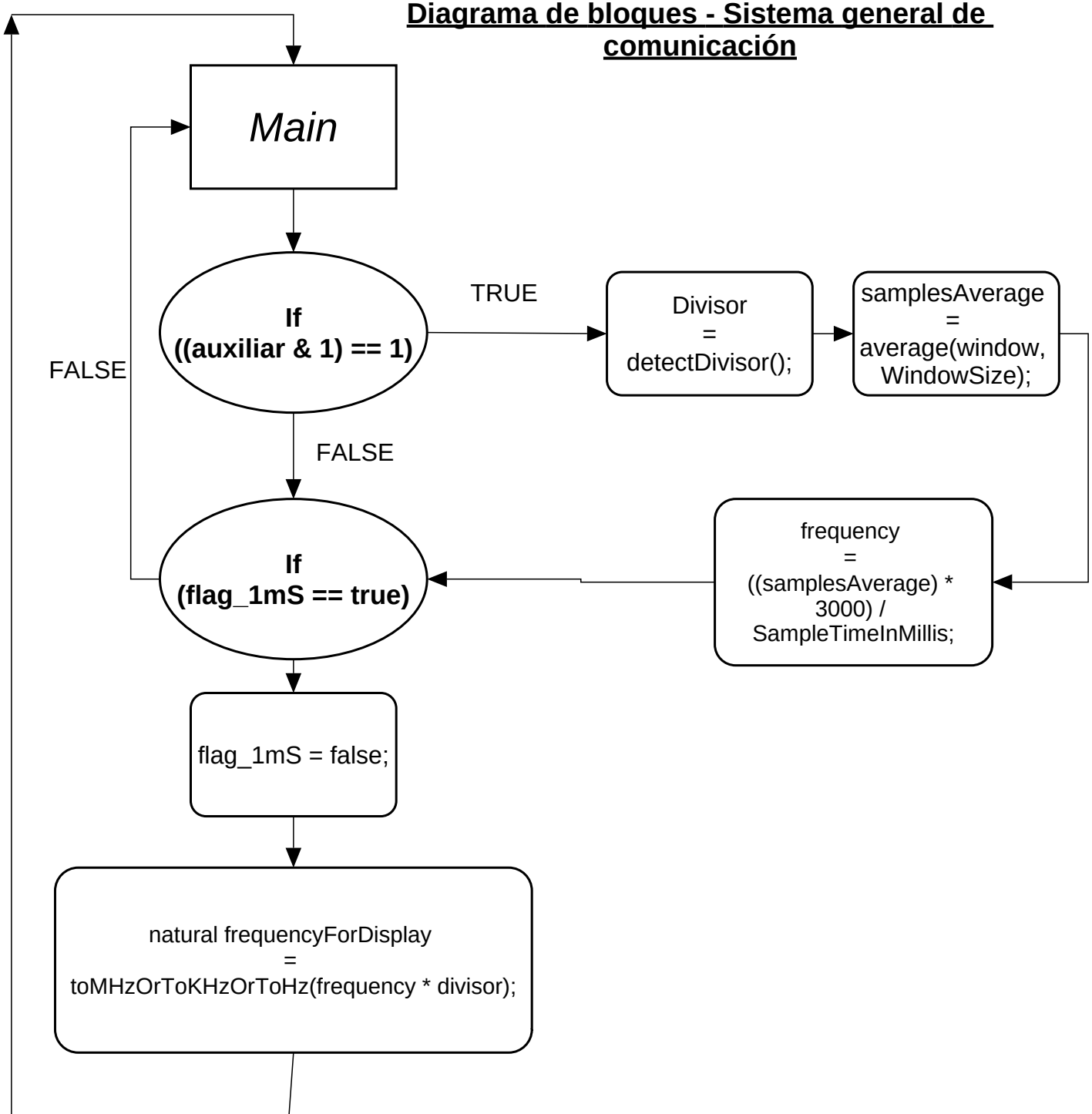
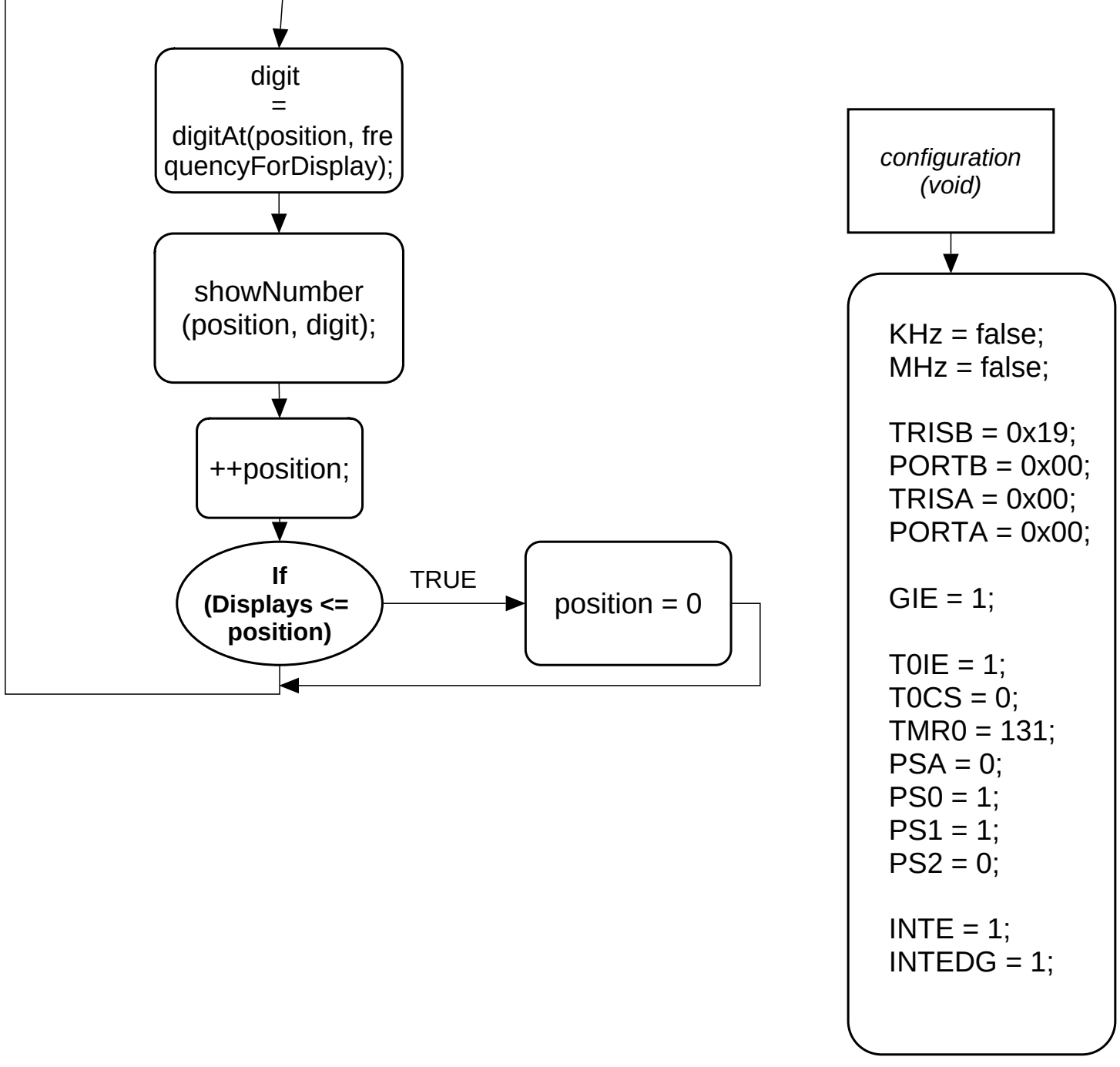
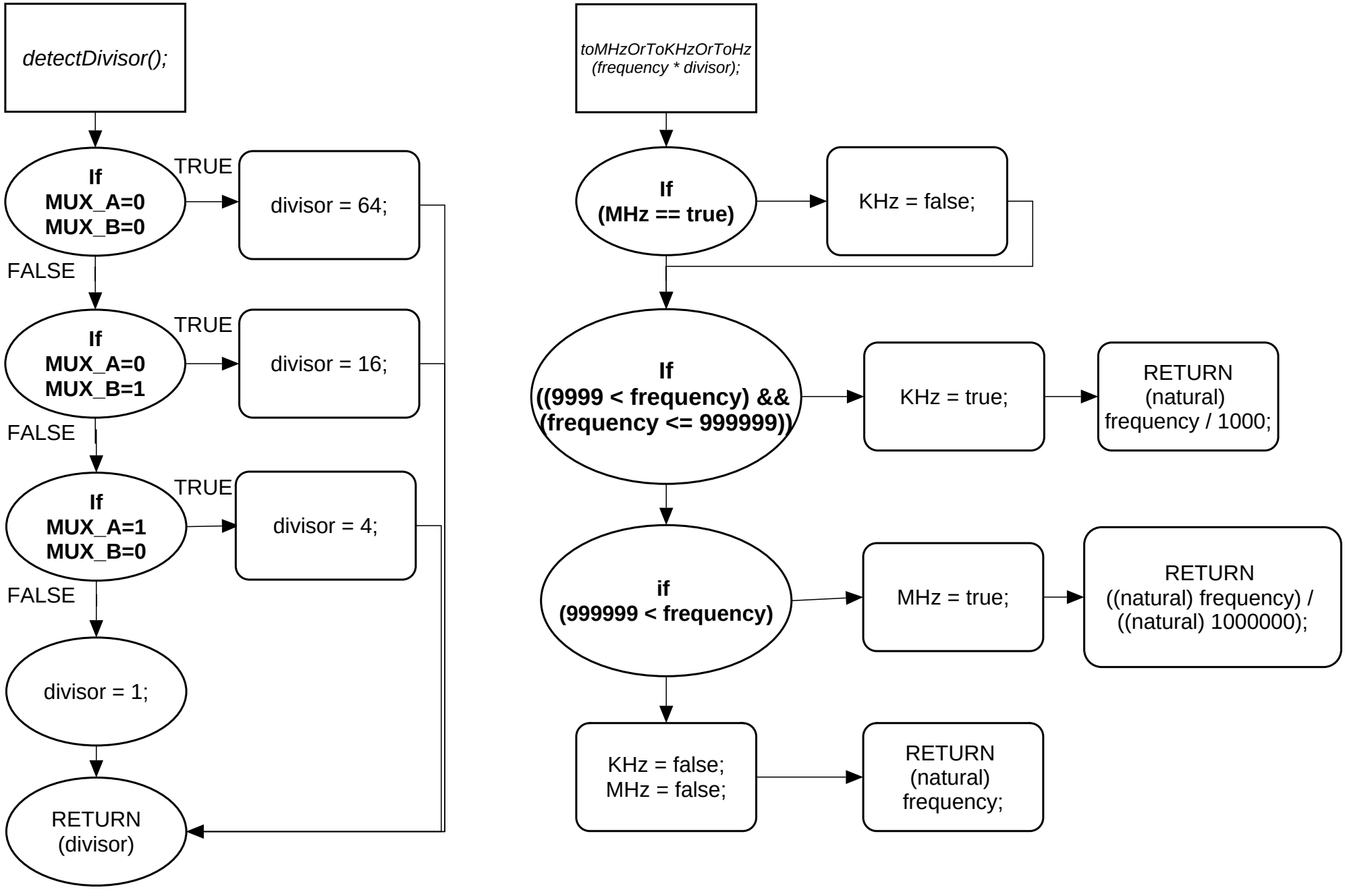
Diagrama de bloques - Sistema general de comunicación

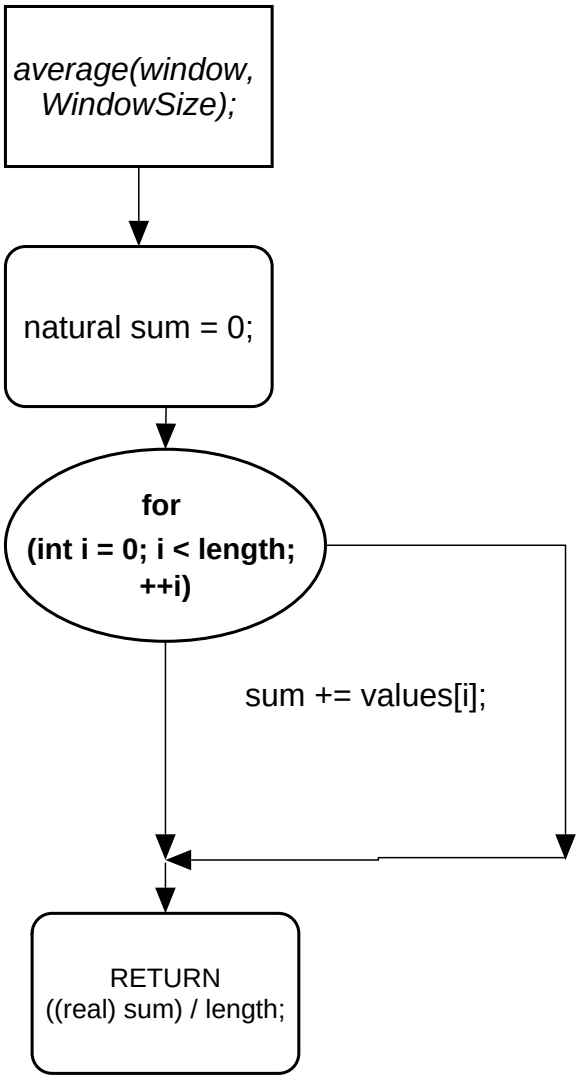
```
// Definición de variables:
typedef __uint24 natural;
typedef float real;
```

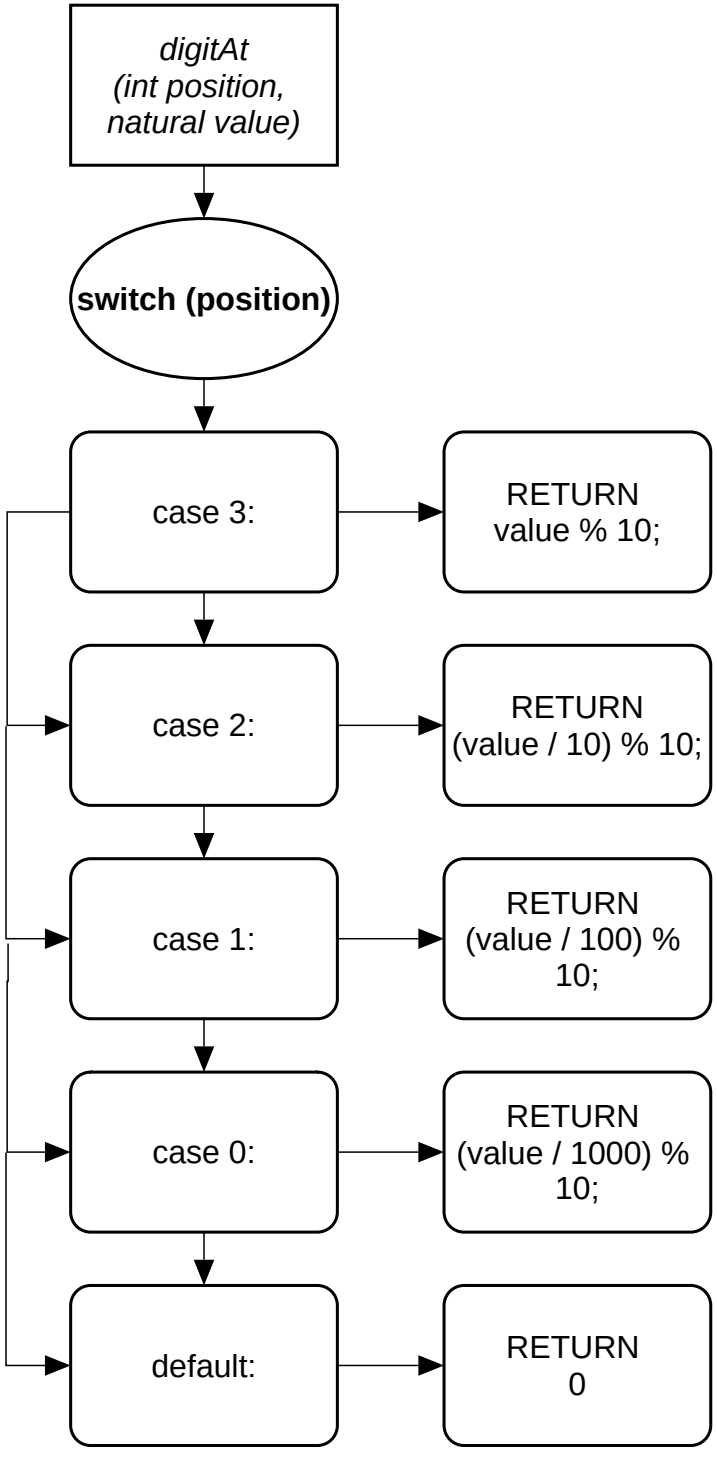
```
// Variables Globales:
natural flankChanges = 0;
natural window[WindowSize];

int timerTicks = 0;
int sampleTicks  = 0;
bool flag_1mS = false;
int auxiliar = 0;
```

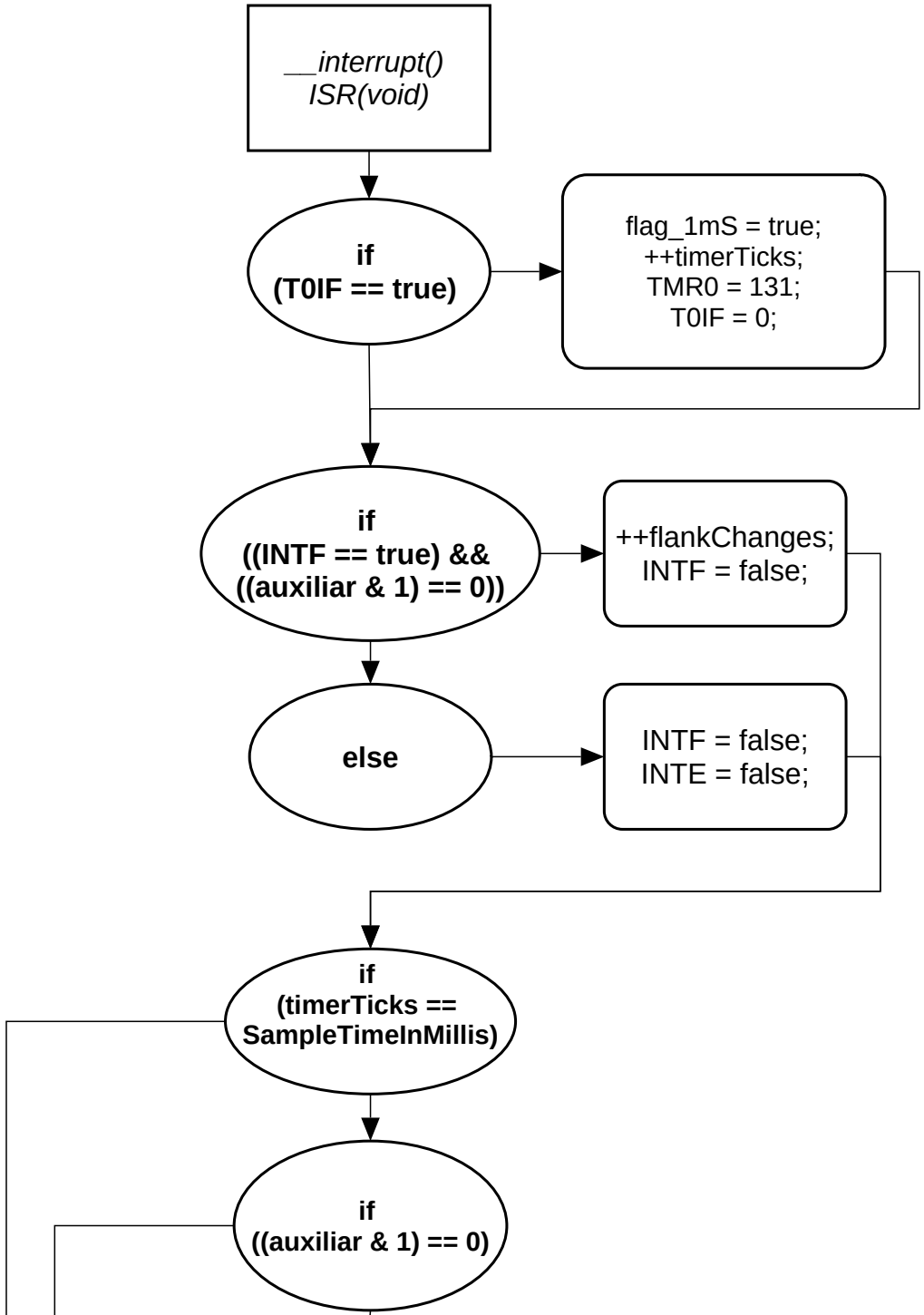


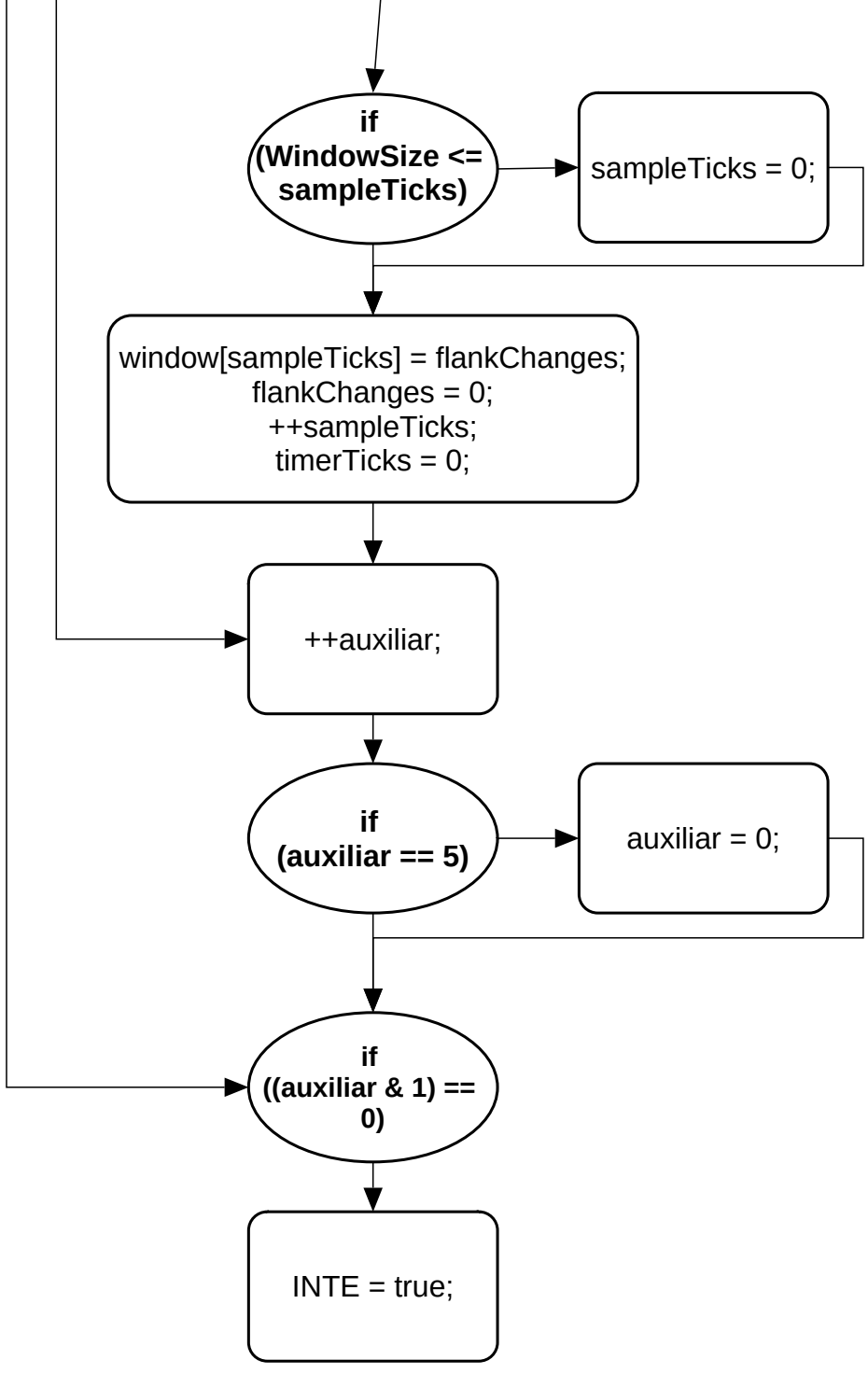

digit
=
digitAt(position, frequencyForDisplay);
showNumber
(position, digit);
++position;
If
(Displays <=
position)
TRUE
position = 0
configuration
(void)
KHz = false;
MHz = false;
TRISB = 0x19;
PORTB = 0x00;
TRISA = 0x00;
PORTA = 0x00;
GIE = 1;
T0IE = 1;
T0CS = 0;
TMR0 = 131;
PSA = 0;
PS0 = 1;
PS1 = 1;
PS2 = 0;
INTE = 1;
INTEDG = 1;

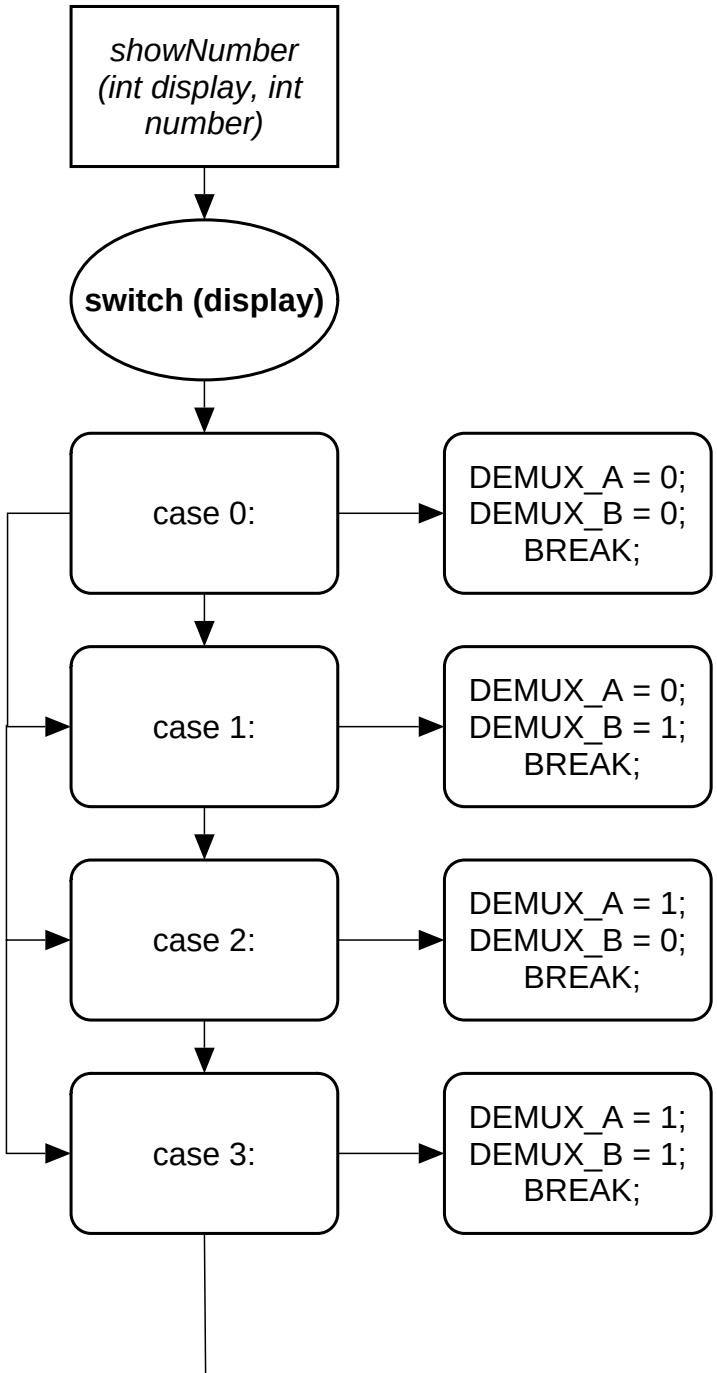

detectDivisor();
If
MUX_A=0
MUX_B=0
TRUE
divisor = 64;
FALSE
If
MUX_A=0
MUX_B=1
TRUE
divisor = 16;
FALSE
If
MUX_A=1
MUX_B=0
TRUE
divisor = 4;
FALSE
divisor = 1;
RETURN
(divisor)
toMHzOrToKHzOrToHz
(frequency * divisor);
If
(MHz == true)
KHz = false;
If
((9999 < frequency) &&
(frequency <= 999999))
KHz = true;
RETURN
(natural)
frequency / 1000;
if
(999999 < frequency)
MHz = true;
RETURN
((natural) frequency) /
((natural) 1000000);
KHz = false;
MHz = false;
RETURN
(natural)
frequency;

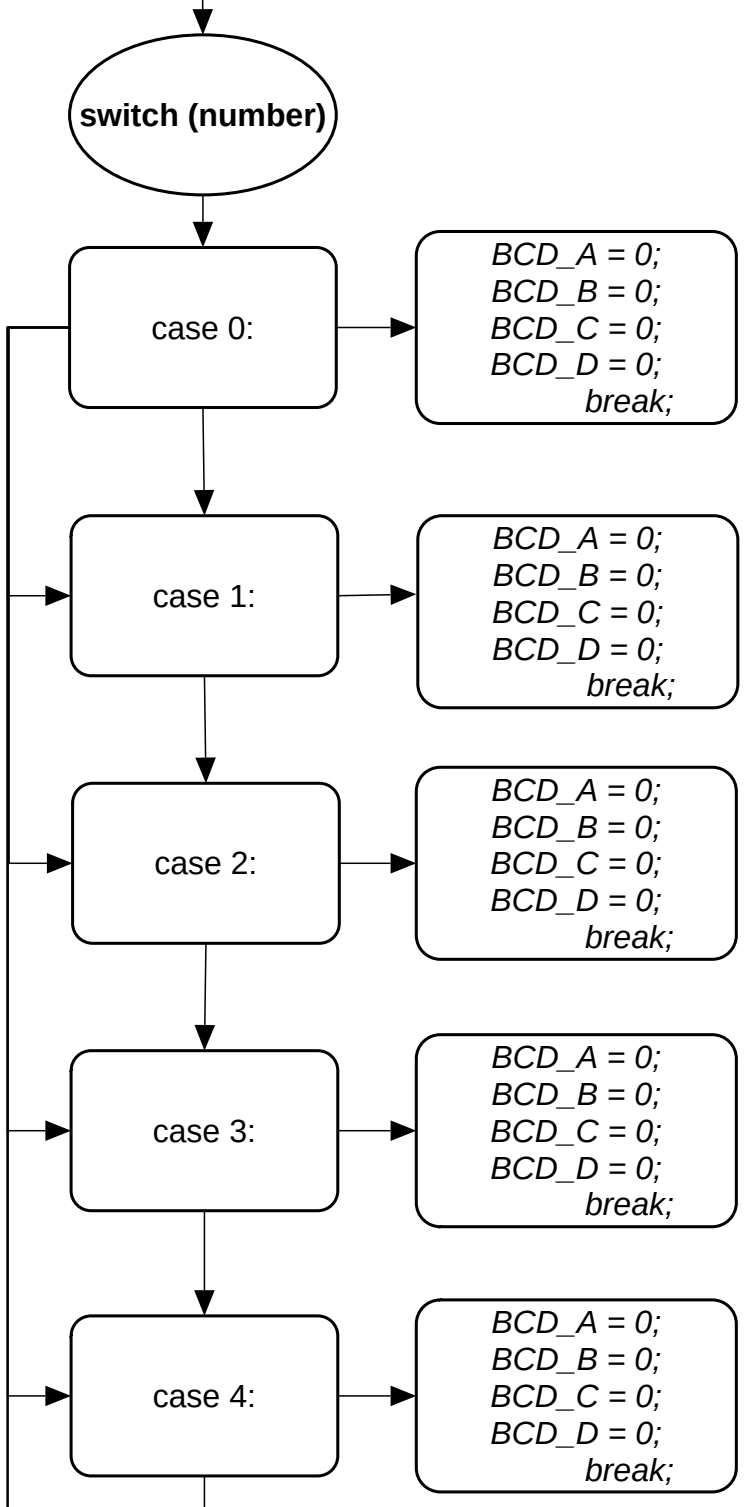

average(window, WindowSize);
natural sum = 0;
for (int i = 0; i < length; ++i)
sum += values[i];
RETURN ((real) sum) / length;

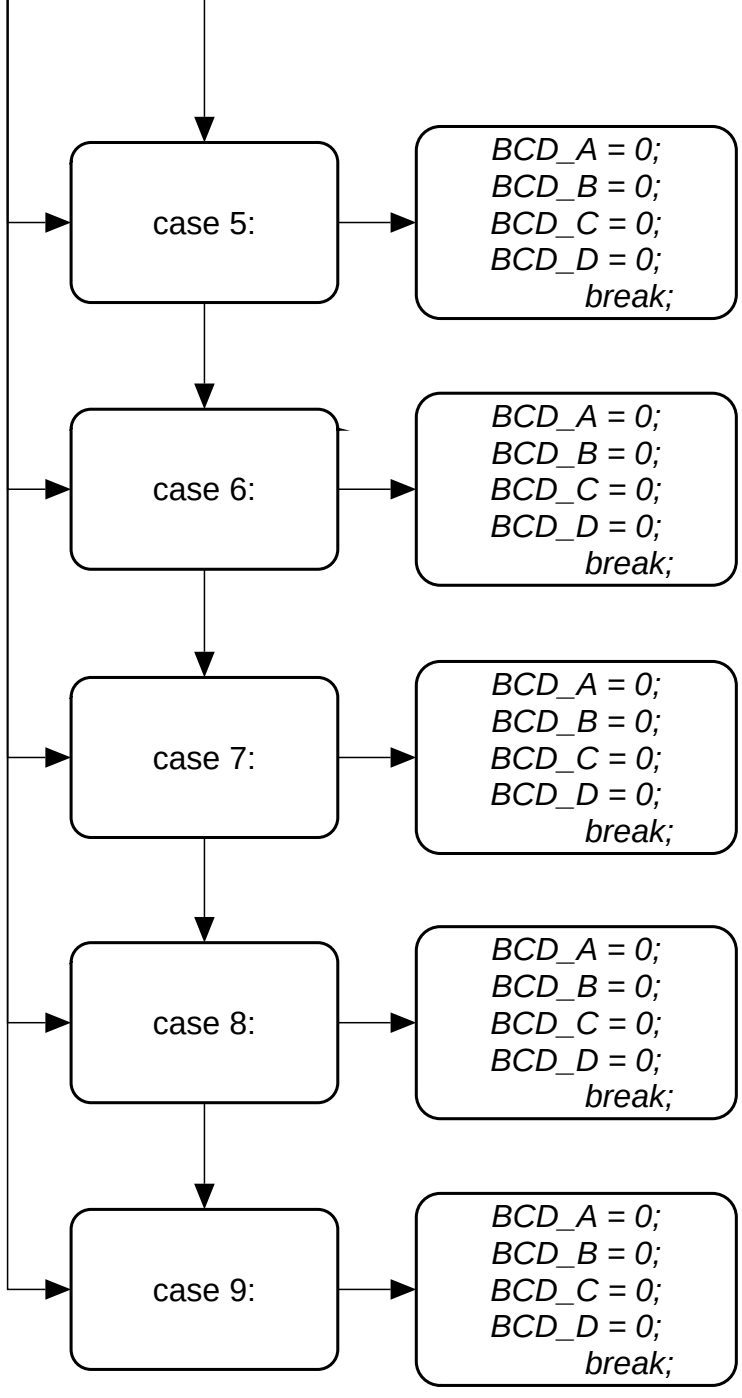

digitAt (int position, natural value)
switch (position)
case 3:
RETURN value % 10;
case 2:
RETURN (value / 10) % 10;
case 1:
RETURN (value / 100) % 10;
case 0:
RETURN (value / 1000) % 10;
default:
RETURN 0


__interrupt()
ISR(void)
if
(T0IF == true)
flag_1mS = true;
++timerTicks;
TMR0 = 131;
T0IF = 0;
if
((INTF == true) &&
((auxiliar & 1) == 0))
++flankChanges;
INTF = false;
else
INTF = false;
INTE = false;
if
(timerTicks ==
SampleTimeInMillis)
if
((auxiliar & 1) == 0)


if
(WindowSize <=
sampleTicks)
sampleTicks = 0;
window[sampleTicks] = flankChanges;
flankChanges = 0;
++sampleTicks;
timerTicks = 0;
++auxiliar;
if
(auxiliar == 5)
auxiliar = 0;
if
((auxiliar & 1) ==
0)
INTE = true;


showNumber (int display, int number)
switch (display)
case 0:
DEMUX_A = 0;
DEMUX_B = 0;
BREAK;
case 1:
DEMUX_A = 0;
DEMUX_B = 1;
BREAK;
case 2:
DEMUX_A = 1;
DEMUX_B = 0;
BREAK;
case 3:
DEMUX_A = 1;
DEMUX_B = 1;
BREAK;


switch (number)
case 0:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 1:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 2:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 3:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 4:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;


case 5:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 6:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 7:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 8:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;
case 9:
BCD_A = 0;
BCD_B = 0;
BCD_C = 0;
BCD_D = 0;
break;