**Manual de primitivas de la Biblioteca de Infotronic**

Listas de archivos de las primitivas

**Teclado**

GetKey()

uint8\_t GetKey (void )

Trae del buffer de teclado el codigo de la tecla pulsada.

Parametros

void

Retoro

Codigo de tecla o NO\_KEY (0xFF)

**LCD**

LCD\_Display()

void LCD\_Display (const char \*string, uint8\_t line, uint8\_t pos)

Muestra una srting en un LCD de 2 x 16.

Parametros

sring direcion de comienzo de la string

line numero de renglon del Display

Parametros de line

DSP0: renglon superior

DSP1: renglon inferior

pos posicion relativa dentro del renglon

Retorno

void

**Relays**

Relays()

void Relays (uint8\_t nRelay, uint8\_t estado)

Activa o desactiva un relay.

Parametros

numero de relay

modo Selecciona entre los 4 modos de funcionamiento

Numeros de Relays en las siguientes macros

RELAY0: 0

RELAY1: 1

RELAY2: 2

RELAY3: 3

Retorno

void

[**RGB**](file:///C:\Users\socco\Documents\MCUXpressoIDE_10.1.1_606\workspace\bibliotecaInfoII\doc\html\_p_r___r_g_b_8c.html)

LedsRGB()

void LedsRGB (uint8\_t led, uint8\_t estado)

Activa o desactiva uno de los leds del RGB.

Parametros

Color de led

los colores estan en las siguientes macros

ROJO: 0

VERDE: 1

AZUL: 2

estado indica si apaga o prende

Retorno

void

**Entradas**

Entradas()

uint8\_t Entradas (uint8\_t nEntrada)

Valor de una de las entradas digitales.

Parametros

uint8\_t numero de entrada

Numeros de entrada en las siguientes macros

ENTRADA0 : 0

ENTRADA1 : 1

ENTRADA2 : 2

Retorno

uint8\_t valor de la entrada pasada como parámetro

**7seg**

Display()

void Display (unsigned int Val, unsigned char dsp)

Primitiva de display 7 segmentos.

Muestra el valor que recibe en Val en el display indicado por dsp, los seis digitos de Infotronic constituyen dos displays de 3 digitos

Parametros

unsigned int Val: Valor a mostar en el display elegido

unsigned char dsp: Display elegido para mostar el valor Val

Retorno

void

**Timers**

TimerStart

void TimerStart(uint8\_t event, timer\_t t, void (\*handler)(void))

Inicia un timer

Inicia el timer \a e al transcurrir el tiempo especificado por \a t se llama a la funcion apuntada por \a handler

Parametros

event: Numero de evento entre 0 y 31

t: Tiempo del evento. Dependiente de la base de tiempos

handler: Callback del evento

Retorno

void

SetTimer

void SetTimer( uint8\_t event , timer\_t t )

Inicia un timer

Reinicia el timer con el valor t (no lo resetea)

Parametros

event: Numero de evento entre 0 y 31

t: Tiempo del evento. Dependiente de la base de tiempos

Retorno

void

GetTimer

uint32\_t GetTimer( uint8\_t event )

Toma el valor al vuelo del timer en cuestion

Lee el timer

Parametros

event: Numero de evento entre 0 y 31

Retorno

uint32\_t valor del timer

StandByTimer

void StandByTimer( uint8\_t event , uint8\_t accion)

Detiene/Arranca el timer, NO lo resetea

lo pone o lo saca de stand by

Parametros

event: Numero de evento entre 0 y 31

accion: RUN lo arranca, PAUSE lo pone en stand by

Retorno

void

Timer\_Stop

void Timer\_Stop(Eventos e)

Detiene un timer

details Detiene el timer e

Parametros

e: Numero de evento entre 0 y 31

Retorno

void

Timer\_Close

Timer\_Close(void)

Detiene los timers

Detiene todos los timers

Parametros

void

Retorno

void

**UART**

Transmitir

int16\_t Transmitir ( uint8\_t com , const void \* datos , uint8\_t cant)

Despacha los datos a transmitir

Parametros

uint8\_t com: Puerto que sera utilizado

const void \*: datos: datos a transmitir

uint8\_t cant: cantidad de datos a transmitir

Retorno

0 por exito, -1 por Error (datos exedidos)

**Adc**

Temperatura

int16\_t Temperatura ( void )

Retorna el valor de temperatura del termistor

Parametros

void

Retorno

temperatura (rango -50 a 120)

Potenciometro

int16\_t Potenciometro ( void )

Retorna el valor de tension asociado al potenciometro

Parametros

void

Retorno

Lectura del pote (rango 0 a 3.3V )

ADC\_Externa

int16\_t ADC\_Externa ( void )

Retorna el valor de tension asociado al potenciometro

Parametro

void

Retorno

Lectura de cuentas

ADC\_Promedio

int16\_t ADC\_Promedio(uint8\_t canal)

Retorna el valor promedio de MAX\_MUESTRAS asociado a un canal

Parametro

uint8\_t canal: Canal a medir

Retorno

Lectura de entrada analogica 2