Flowcode V5 CAL (Code Abstraction Layer) Documentation

CAL Contents

ADC CAL

ADC Functions

void FC_CAL_Enable_ADC (MX_UINT8 Channel, MX_UINT8 Conv_Speed, MX_UINT8 Vref, MX_UINT8 T_Charge);

MX_UINT16 FC_CAL_Sample_ADC (MX_UINT8 Sample_Mode);

void FC_CAL_Disable_ADC ();

Enable Definition

MX_ADC_REF

CAN CAL

Internal CAN Functions

void FC_CAL_Internal_CAN_Init (MX_UINT8 ref);

void FC_CAL_Internal_CAN_Uninit (MX_UINT8 ref);

void FC_CAL_Internal_CAN_Send_Buffer (MX_UINT8 ref, MX_UINT8 buffer);

MX_UINT8 FC_CAL_Internal_CAN_CheckRX (MX_UINT8 ref, MX_UINT8 buffer);

void FC_CAL_Internal_CAN_Set_TX_Data(MX_UINT8 ref, MX_UINT8 buffer, MX_UINT8 data_cnt, MX_UINT8 d0, MX_UINT8 d1, MX_UINT8 d2, MX_UINT8 d3, MX_UINT8 d4, MX_UINT8 d5, MX_UINT8 d6, MX_UINT8 d7);

void FC_CAL_Internal_CAN_Set_TX_Std_ID(MX_UINT8 ref, MX_UINT8 buffer, MX_UINT8 hi, MX_UINT8 lo);

void FC_CAL_Internal_CAN_Set_TX_Exd_ID(MX_UINT8 ref, MX_UINT8 buffer, MX_UINT8 id3, MX_UINT8 id2, MX_UINT8 id1, MX_UINT8 id0);

void FC_CAL_Internal_CAN_Set_Std_RX_Filter(MX_UINT8 ref, MX_UINT8 filter, MX_UINT8 hi, MX_UINT8 lo);

void FC_CAL_Internal_CAN_Set_Exd_RX_Filter(MX_UINT8 ref, MX_UINT8 filter, MX_UINT8 id3, MX_UINT8 id2, MX_UINT8 id1, MX_UINT8 id0);

void FC_CAL_Internal_CAN_Set_Std_RX_Mask(MX_UINT8 ref, MX_UINT8 filter, MX_UINT8 hi, MX_UINT8 lo);

void FC_CAL_Internal_CAN_Set_Exd_RX_Mask(MX_UINT8 ref, MX_UINT8 filter, MX_UINT8 id3, MX_UINT8 id2, MX_UINT8 id1, MX_UINT8 id0);

CAN Definitions

MX_CAN_CHANNEL_X

MX_CAN_BAUD1_X

MX_CAN_BAUD2_X

MX_CAN_BAUD3_X

Enable Definition

MX_CAN_REF1

MX_CAN_REF2

Cap Touch CAL

Cap Touch Functions

void FC_CAL_CAP_Touch_Init (void)

void FC_CAL_CAP_Touch_Reset (void)

void FC_CAL_CAP_Touch_Drain (void)

void FC_CAL_CAP_Touch_Charge (void)

void FC_CAL_CAP_Touch_Stop (void)

Enable Definition

MX_CAPTOUCH_REF

Delay CAL

```
void Wdt_Delay_S(char delay);
void Wdt_Delay_Ms(char delay);
void delay_10us(char del);
void delay_us(char del);
void delay_ms(char del);
void delay_s(char del);
void nop(void);
```

Auto Enabled

EEPROM CAL

EEPROM Function Prototypes

MX_UINT8 FC_CAL_EE_Read (MX_UINT16 Address);

void FC_CAL_EE_Write (MX_UINT16 Address, MX_UINT8 Data);

Enable Definition

MX_EEPROM_REF

Float CAL

```
flt_add(a, b)                    // Binary arithmetic
flt_sub(a, b)
flt_mul(a, b)
flt_div(a, b)
flt_rem(a, b)
flt_neg(a)                       //Unary arithmetic
flt_eq(a, b)                     //Comparisons
flt_ne(a, b)
flt_ge(a, b)
flt_gt(a, b)
flt_le(a, b)
flt_lt(a, b)
flt_toi(a)                       //Assignment
flt_fromi(a)
char isinf(float f)
```

Auto Enabled

I2C CAL

I2C Master Functions

void FC_CAL_I2C_Master_Init (MX_UINT8 ref);

void FC_CAL_I2C_Master_Uninit (MX_UINT8 ref);

void FC_CAL_I2C_Master_Start (MX_UINT8 ref);

void FC_CAL_I2C_Master_Restart (MX_UINT8 ref);

void FC_CAL_I2C_Master_Stop (MX_UINT8 ref);

MX_UINT8 FC_CAL_I2C_Master_TX_Byte (MX_UINT8 ref, MX_UINT8 Data);

MX_UINT8 FC_CAL_I2C_Master_RX_Byte (MX_UINT8 ref, MX_UINT8 Last);

Slave functions currently not implemented

void FC_CAL_I2C_Slave_Init (MX_UINT8 ref);

void FC_CAL_I2C_Slave_Uninit (MX_UINT8 ref);

void FC_CAL_I2C_Slave_SetTxData (MX_UINT8 ref, Data);

MX_UINT8 FC_CAL_I2C_Slave_GetRxData (MX_UINT8 ref, MX_UINT8 Last);

I2C Control Definitions

MX_I2C_CHANNEL_X

MX_I2C_SDA_PIN_X

MX_I2C_SDA_PORT_X

MX_I2C_SCL_PIN_X

MX_I2C_SCL_PORT_X

MX_I2C_BMODE_X

MX_I2C_BAUD_X

MX_I2C_INT_X

Enable Definition

MX_I2C_REF1

MX_I2C_REF2

MX_I2C_REF3

MX_I2C_REF4

ICD CAL

ICD Functions

void FC_MARKER(signed int block_id);

void ICD_PUSH(char macro_id);

void ICD_POP();

unsigned char icd_byte_tx(char data);

unsigned char icd_byte_rx(void);

unsigned char wait_clock(char state, char timeout_inc);

unsigned char wait_data(char state);

char read_register(unsigned int address);

void write_register(unsigned int address, char data);

void rst_dev(void);

Enable Definition

USE_FLOWCODE_ICD

I/O CAL

MX_UINT8 FC_CAL_Bit_In(MX_UINT8 prt, MX_UINT8 bt);

MX_UINT8 FC_CAL_Bit_In_DDR(MX_UINT8 prt, MX_UINT8 trs, MX_UINT8 bt);

void FC_CAL_Bit_High(MX_UINT8 prt, MX_UINT8 bt);

void FC_CAL_Bit_High_DDR(MX_UINT8 prt, MX_UINT8 trs, MX_UINT8 bt);

void FC_CAL_Bit_Low(MX_UINT8 prt, MX_UINT8 bt);

void FC_CAL_Bit_Low_DDR(MX_UINT8 prt, MX_UINT8 trs, MX_UINT8 bt);

Auto Enabled

PWM CAL

I2C Master Functions

void FC_CAL_PWM_Enable_Channel (MX_UINT8 ref, MX_UINT8 tmr_prph);

void FC_CAL_PWM_Disable_Channel (MX_UINT8 ref);

void FC_CAL_PWM_Set_Duty_8Bit (MX_UINT8 ref, MX_UINT8 duty);

void FC_CAL_PWM_Change_Period (MX_UINT8 ref, MX_UINT8 period, MX_UINT16 prescaler);

void FC_CAL_PWM_Set_Duty_10Bit (MX_UINT8 ref, MX_UINT16 duty);

PWM Control Definitions

MX_PWM_CHANNEL_X

MX_PWM_HWALT_X

Enable Definition

MX_PWM_REF1

MX_PWM_REF2

MX_PWM_REF3

MX_PWM_REF4

MX_PWM_REF5

MX_PWM_REF6

MX_PWM_REF7

MX_PWM_REF8

MX_PWM_REF9

MX_PWM_REF10

SPI CAL

SPI Master Functions

void FC_CAL_SPI_Master_Init (MX_UINT8 ref);

void FC_CAL_SPI_Master_Uninit (MX_UINT8 ref);

MX_UINT8 FC_CAL_SPI_Master_Byte (MX_UINT8 ref, MX_UINT8 DataOut);

//Slave functions currently not implemented

void FC_CAL_SPI_Slave_Init (MX_UINT8 ref);

void FC_CAL_SPI_Slave_Uninit (MX_UINT8 ref);

void FC_CAL_SPI_Slave_SetTxData (MX_UINT8 ref, Data);

MX_UINT8 FC_CAL_SPI_Slave_GetRxData (MX_UINT8 ref);

SPI Control Definitions

MX_SPI_CHANNEL_X

MX_SPI_MOSI_PIN_X

MX_SPI_MOSI_PORT_X

MX_SPI_MISO_PIN_X

MX_SPI_MISO_PORT_X

MX_SPI_SCK_PIN_X

MX_SPI_SCK_PORT_X

MX_SPI_CS_PIN_X

MX_SPI_CS_PORT_X

MX_SPI_BMODE_X

MX_SPI_PRESCALE_X

MX_SPI_POSTSCALE_X

MX_SPI_INT_X

Enable Definition

MX_SPI_REF1

MX_SPI_REF2

MX_SPI_REF3

MX_SPI_REF4

String CAL

char FCI_GETCHAR(char* sStr1, char iStr1_len, char iPos);

char FCI_GETLENGTH(char* sStr1, char iStr1_len);

char FCI_MIDSTRING(char* sSrc, char iSrc_len, char* sDst, char iDst_len, char iStart, char iCount);

char FCI_LEFTSTRING(char* sSrc, char iSrc_len, char* sDst, char iDst_len, char iCount);

char FCI_RIGHTSTRING(char* sSrc, char iSrc_len, char* sDst, char iDst_len, char iCount);

char FCI_CONCATENATE(char* sSrc1, char iSrc1_len, char* sSrc2, char iSrc2_len, char* sDst, char iDst_len);

char FCI_TOSTRING(int iSrc1, char* sDst, char iDst_len);

void FCI_TOLOWER(char* sSrc, char iSrc_len, char* sDst, char iDst_len);

void FCI_TOUPPER(char* sSrc, char iSrc_len, char* sDst, char iDst_len);

char FCI_COMPARE(char* sSrc1, char iSrc1_len, char* sSrc2, char iSrc2_len, char iNoCase);

char FCI_FLOAT_TO_STRING(float Number, char Precision, char* String, char MSZ_String);

char FCI_NUMBER_TO_HEX(short Number, char* String, char MSZ_String);

long FCI_STRING_TO_INT(char* String, char MSZ_String);

float FCI_STRING_TO_FLOAT(char* String, char MSZ_String);

char FCI_STRREV(char* sSrc, char iSrc_len, char* sDst, char iDst_len);

char FCI_STRMOV(char* sSrc, char iSrc_len, char* sDst, char iDst_len);

char* FCI_SHEAD(char* sSrc1, char iSrc1_len, char* sSrc2, char iSrc2_len, char* sDst, char iDst_len);

char* FCI_SCOPY(char* sSrc, char iSrc_len, char* sDst, char iDst_len);

Auto Enabled

TypeDefs Variable CAL

MX_BOOL	Boolean	0 or 1
MX_SINT8	Signed integer 8-bit	–128 to 127
MX_SINT16	Signed integer 16-bit	-32768 to 32767
MX_SINT32	Signed integer 32-bit	-2147483648 to 2147483647
MX_UINT8	Unsigned integer 8-bit	0 to 255
MX_UINT16	Unsigned integer 16-bit	0 to 65535
MX_UINT32	Unsigned integer 32-bit	0 to 4294967295
MX_CHAR	Unsigned integer 8-bit	0 to 255
MX_STRING	Unsigned integer 8-bit	0 to 255
MX_FLOAT	Signed float 32-bit	+/- 2 to the power 128
MX_Union32	AsLong	Long Variable
MX_Union32	AsInt[]	Int Variable – 0/1
MX_Union32	AsByte[]	Byte Variable – 0/1/2/3
MX_Union16	AsInt	Int Variable
MX_Union16	AsByte[]	Byte Variable – 0/1

Auto Enabled

UART CAL

Function Prototypes

void FC_CAL_UART_Init(MX_UINT8 ref);

void FC_CAL_UART_Uninit(MX_UINT8 ref);

void FC_CAL_UART_Send(MX_UINT8 ref, MX_UINT8 nChar);

MX_UINT8 FC_CAL_UART_Receive(MX_UINT8 ref, MX_UINT8 nTimeout);

void RS232_Delay(MX_UINT8 ref);

UART Definitions

MX_UART_CHANNEL_X

MX_UART_TX_PIN_X

MX_UART_TX_PORT_X

MX_UART_TX_TRIS_X

MX_UART_RX_PIN_X

MX_UART_RX_PORT_X

MX_UART_RX_TRIS_X

MX_UART_FLOWEN_X

MX_UART_CTS_PIN_X

MX_UART_CTS_PORT_X

MX_UART_CTS_TRIS_X

MX_UART_RTS_PIN_X

MX_UART_RTS_PORT_X

MX_UART_RTS_TRIS_X

MX_UART_DBITS_X

MX_UART_RETURN_X

MX_UART_ECHO_X

MX_UART_INT_X

MX_UART_BAUD_X

MX_SOFT_BAUD_X