



**Processadores Embarcados**

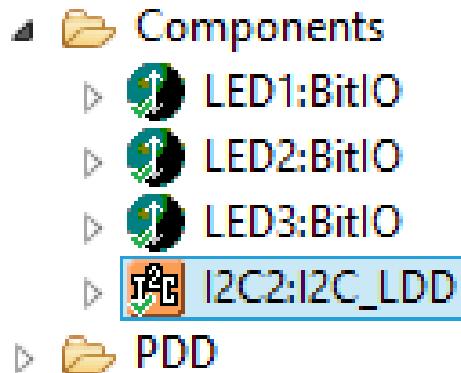
# **Acelerômetro**

## **MMA8451 - I2C**

Profº Fernando Simplicio

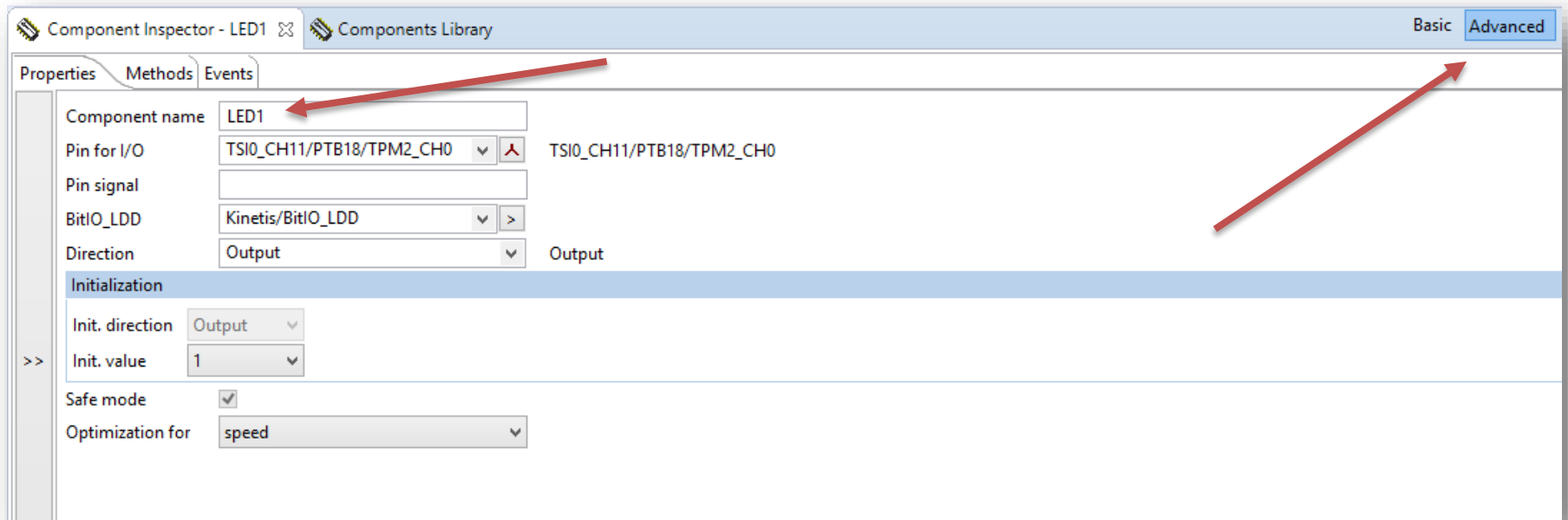
# Acelerômetro

- 1° Crie um Projeto no KDS com os seguintes componentes:



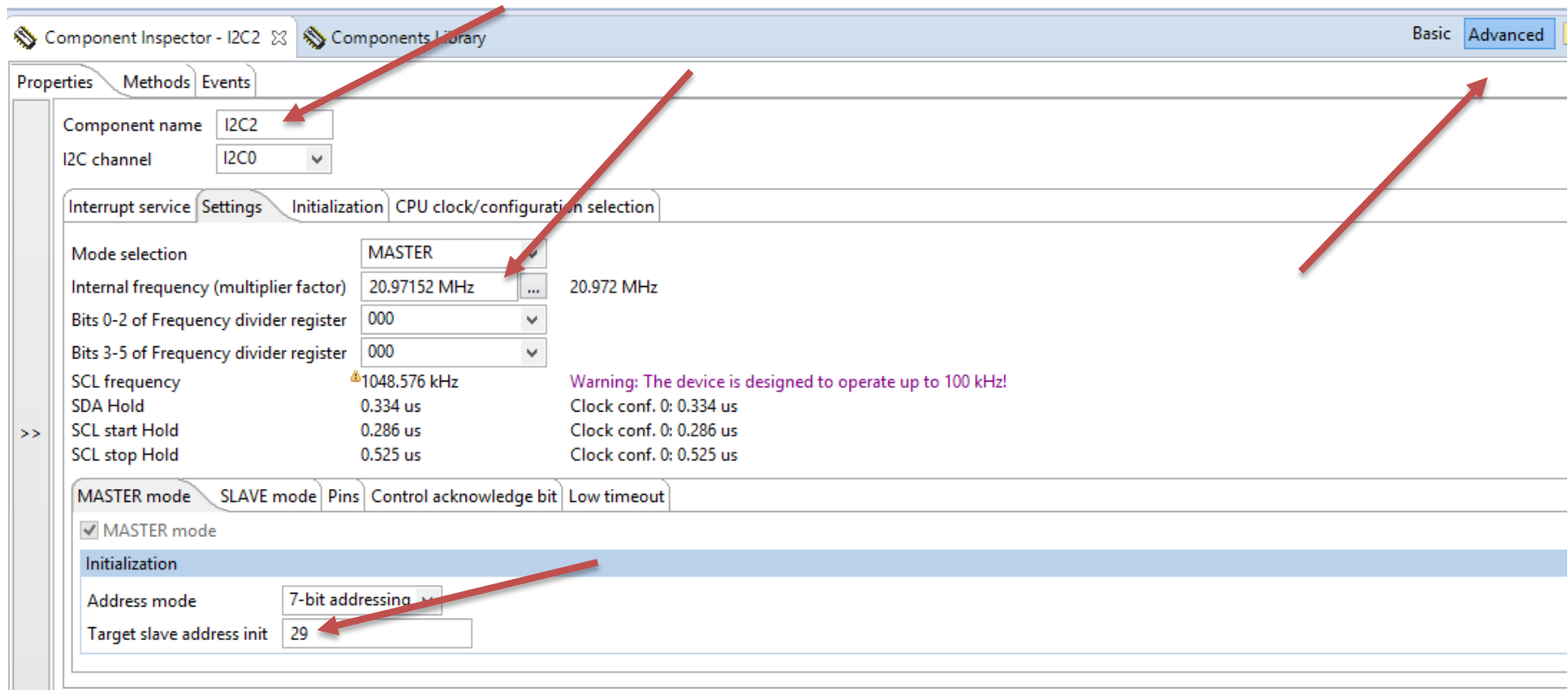
# Acelerômetro

- Altere os nomes dos BitIO.
- Todos os BitIO como saída.



# Acelerômetro

- Configure o Drive I2C conforme apresentado.



# Acelerômetro

## ■ Verifique as configurações dos pinos I2C.

MASTER mode SLAVE mode Pins Control acknowledge bit Low timeout

SDA pin

SDA pin PTE25/TPM0\_CH1/I2C0\_SDA

SDA pin signal

SCL pin

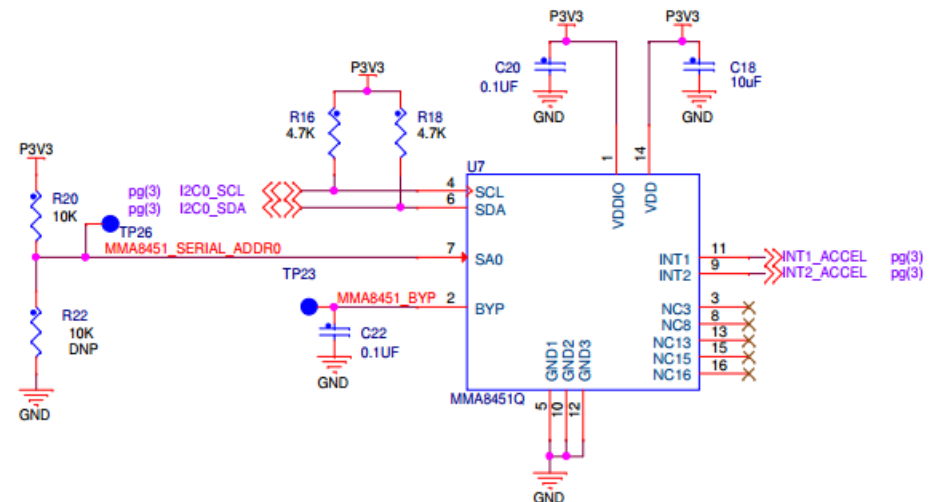
SCL pin PTE24/TPM0\_CH0/I2C0\_SCL

SCL pin signal

High drive select ☐

Input Glitch filter 0

I2C INERTIAL SENSOR



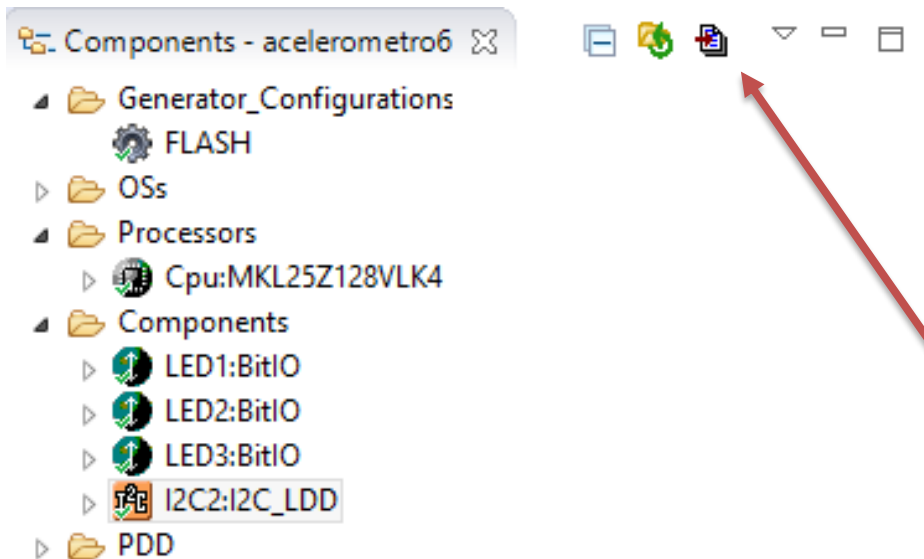
# Acelerômetro

- Habilite a opção “Deinit” na aba “Methods”



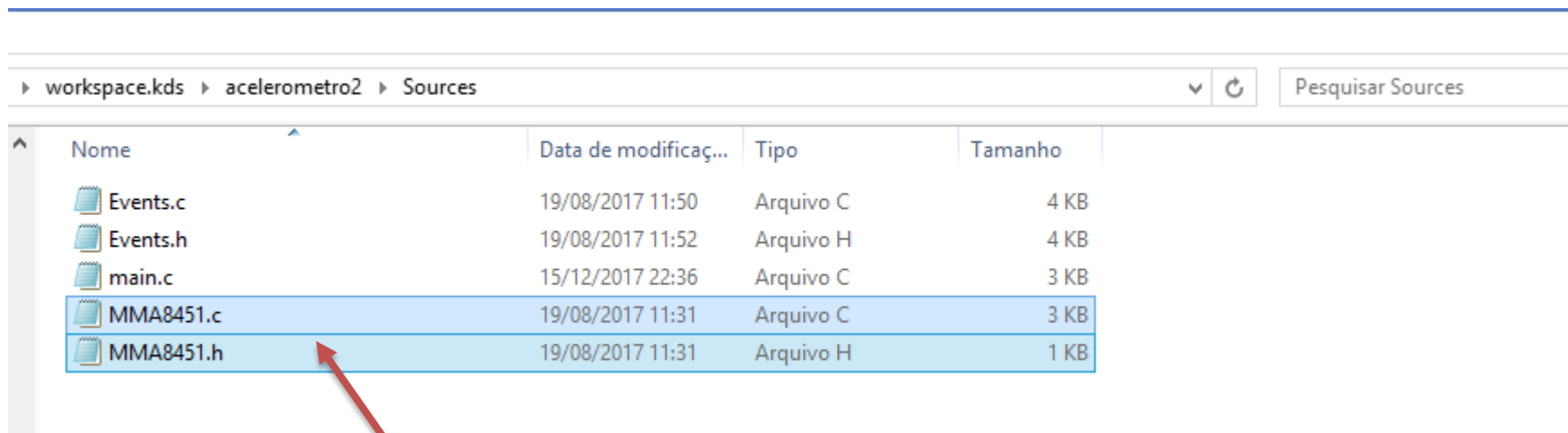
# Acelerômetro

- Clique para gerar os códigos.



# Acelerômetro

- Cole a biblioteca do Acelerômetro dentro da pasta “Sources”.



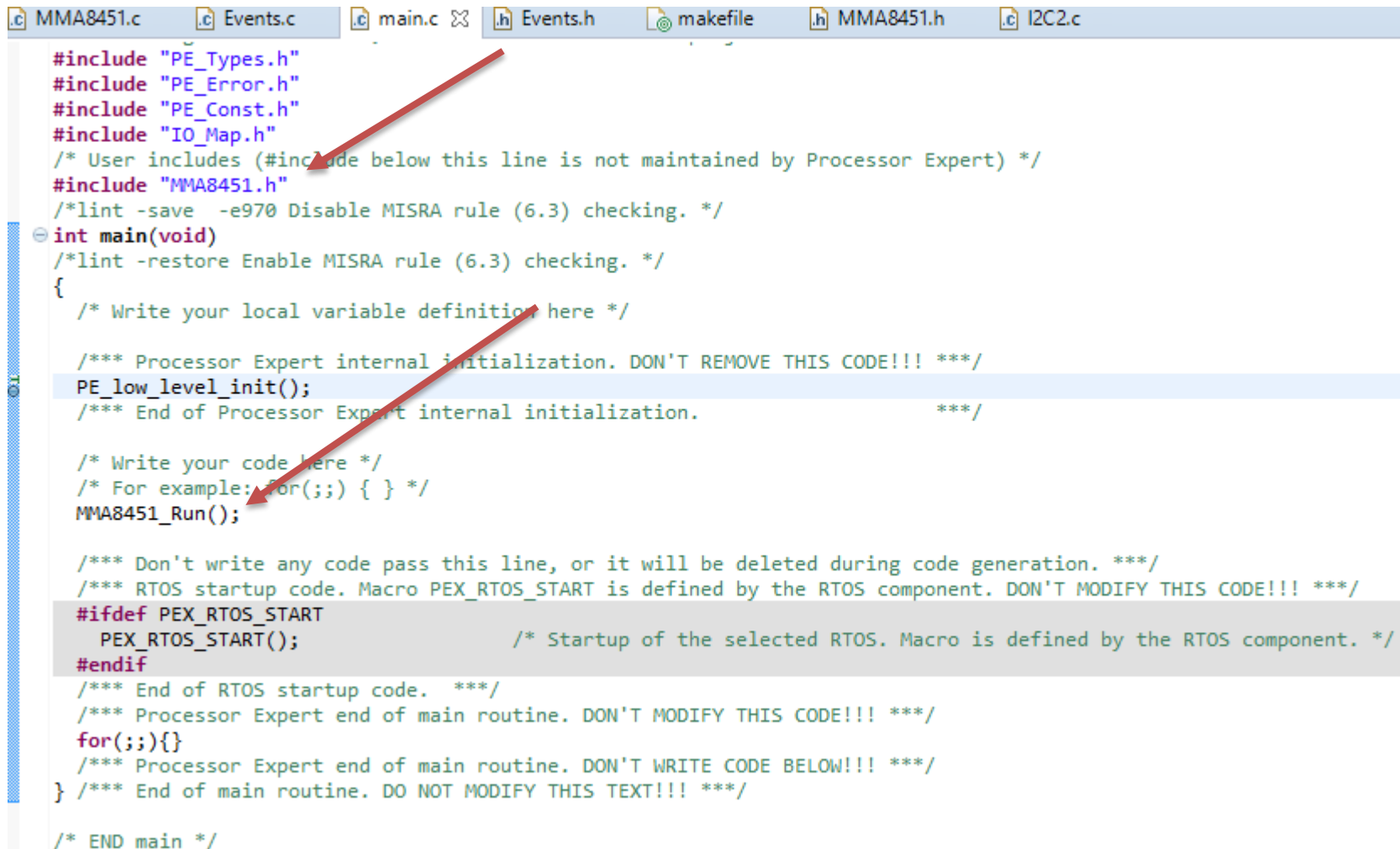
The screenshot shows a file explorer window with the path `workspace.kds > acelerometro2 > Sources`. The table below lists the files in the 'Sources' directory. A red arrow points to the 'MMA8451.h' file.

Nome	Data de modificaç...	Tipo	Tamanho
Events.c	19/08/2017 11:50	Arquivo C	4 KB
Events.h	19/08/2017 11:52	Arquivo H	4 KB
main.c	15/12/2017 22:36	Arquivo C	3 KB
MMA8451.c	19/08/2017 11:31	Arquivo C	3 KB
MMA8451.h	19/08/2017 11:31	Arquivo H	1 KB



# Acelerômetro

## ■ Acrescente em “main.c” ...



```
.c MMA8451.c .c Events.c .c main.c x .h Events.h @ makefile .h MMA8451.h .c I2C2.c

#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
/* User includes (#include below this line is not maintained by Processor Expert) */
#include "MMA8451.h"
/*lint -save -e970 Disable MISRA rule (6.3) checking. */
int main(void)
/*lint -restore Enable MISRA rule (6.3) checking. */
{
    /* Write your local variable definition here */

    /** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! */
    PE_low_level_init();
    /** End of Processor Expert internal initialization. */

    /* Write your code here */
    /* For example: for(;;) { } */
    MMA8451_Run();

    /** Don't write any code pass this line, or it will be deleted during code generation. */
    /** RTOS startup code. Macro PEX_RTOS_START is defined by the RTOS component. DON'T MODIFY THIS CODE!!! */
    #ifdef PEX_RTOS_START
        PEX_RTOS_START(); /* Startup of the selected RTOS. Macro is defined by the RTOS component. */
    #endif
    /** End of RTOS startup code. */
    /** Processor Expert end of main routine. DON'T MODIFY THIS CODE!!! */
    for(;;){}
    /** Processor Expert end of main routine. DON'T WRITE CODE BELOW!!! */
} /** End of main routine. DO NOT MODIFY THIS TEXT!!! */

/* END main */
```

# Acelerômetro

## ■ Acrescente em “events.c” ...

```

MMA8451.c  Events.c  main.c  Events.h  makefile  MMA8451.h  I2C2.c

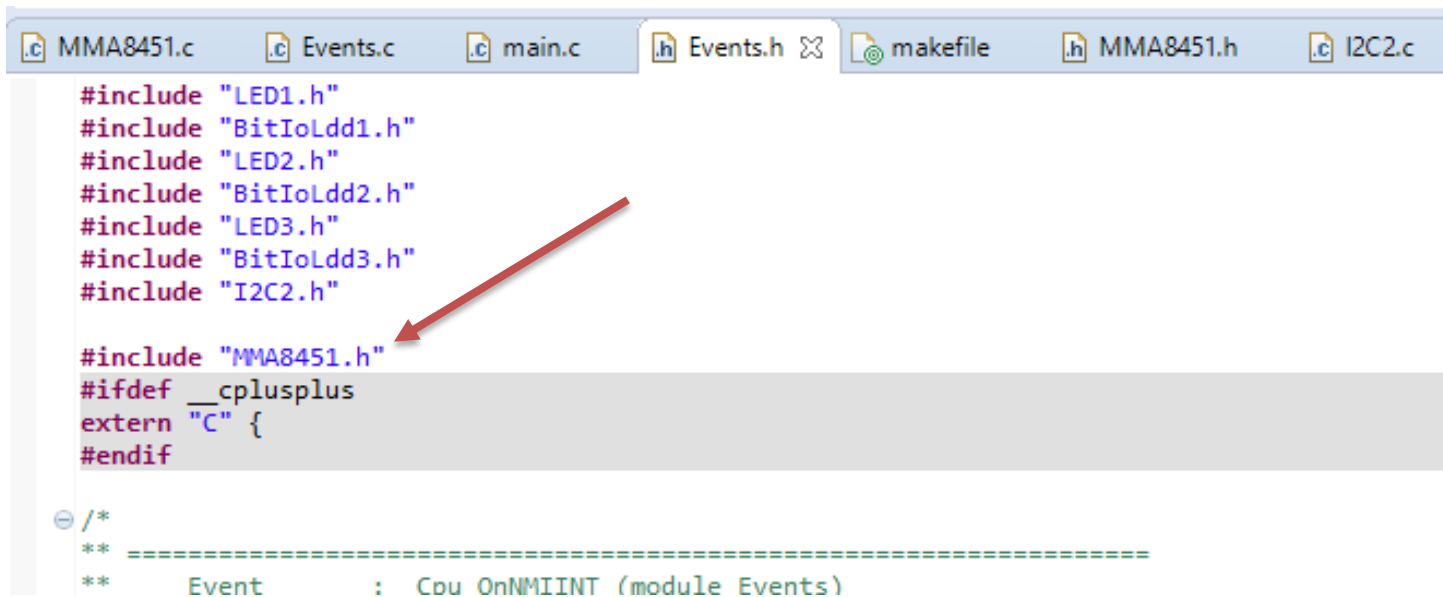
/**
 * as the parameter of Init method.
 */
/* ===== */
void I2C2_OnMasterBlockSent(LDD_UserData *UserDataPtr)
{
    /* Write your code here ... */
    MMA8451_TDataState *ptr = (MMA8451_TDataState*)UserDataPtr;
    ptr->dataTransmittedFlg = TRUE;
}

/* ===== */
/*
 * Event      : I2C2_OnMasterBlockReceived (module Events)
 *
 * Component  : I2C2 [I2C_LDD]
 */
/*!
 * @brief
 * This event is called when I2C is in master mode and finishes
 * the reception of the data successfully. This event is not
 * available for the SLAVE mode and if MasterReceiveBlock is
 * disabled.
 *
 * @param
 * UserDataPtr - Pointer to the user or
 *              RTOS specific data. This pointer is passed
 *              as the parameter of Init method.
 */
/* ===== */
void I2C2_OnMasterBlockReceived(LDD_UserData *UserDataPtr)
{
    /* Write your code here ... */
    MMA8451_TDataState *ptr = (MMA8451_TDataState*)UserDataPtr;
    ptr->dataReceivedFlg = TRUE;
}

```

# Acelerômetro

## ■ Acrescente em “events.h” ...



```
.c MMA8451.c .c Events.c .c main.c .h Events.h makefile .h MMA8451.h .c I2C2.c

#include "LED1.h"
#include "BitIoLdd1.h"
#include "LED2.h"
#include "BitIoLdd2.h"
#include "LED3.h"
#include "BitIoLdd3.h"
#include "I2C2.h"

#include "MMA8451.h"
#ifdef __cplusplus
extern "C" {
#endif

/*
** =====
**      Event      :  Cou OnNMIINT (module Events)
```

# Acelerômetro

**Compile e Grave e veja os resultados!**