SENAI

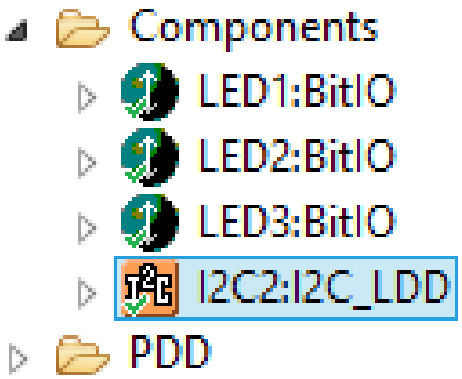
Processadores Embarcados

Acelerômetro MMA8451 - I2C

Prof° Fernando Simplicio

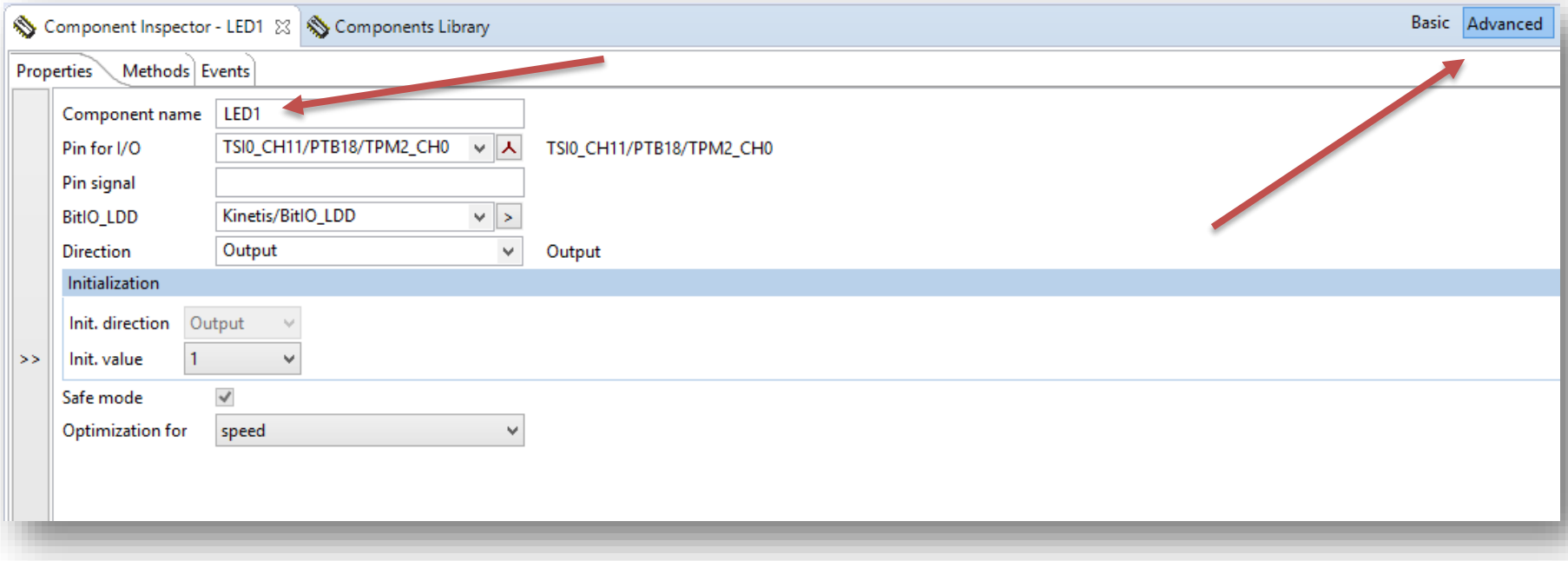
Acelerômetro

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



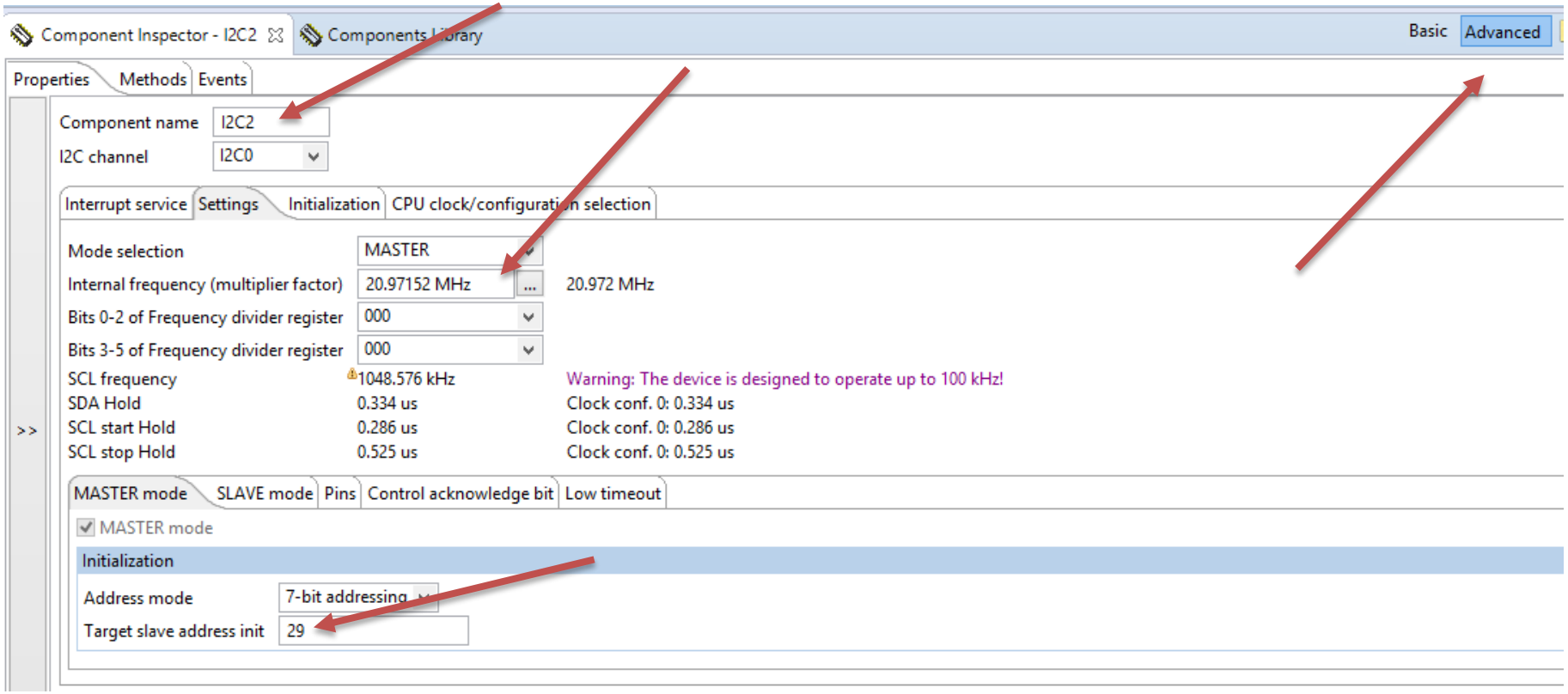
Acelerômetro

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



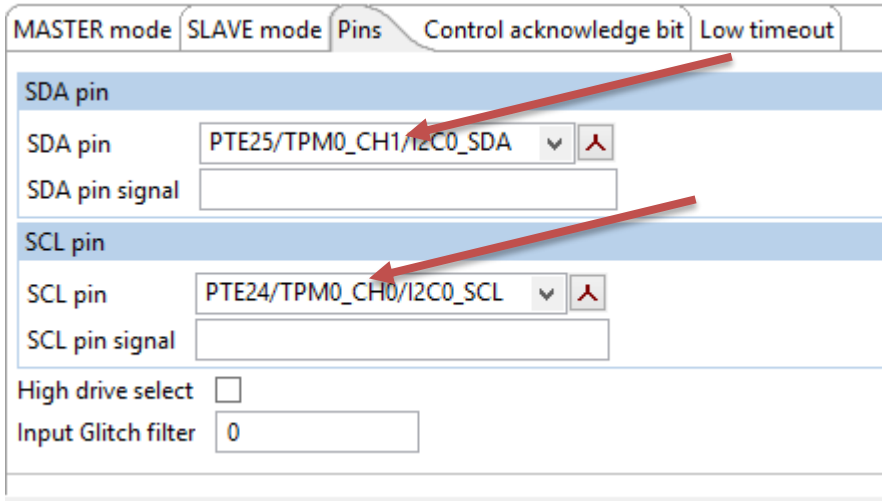
Acelerômetro

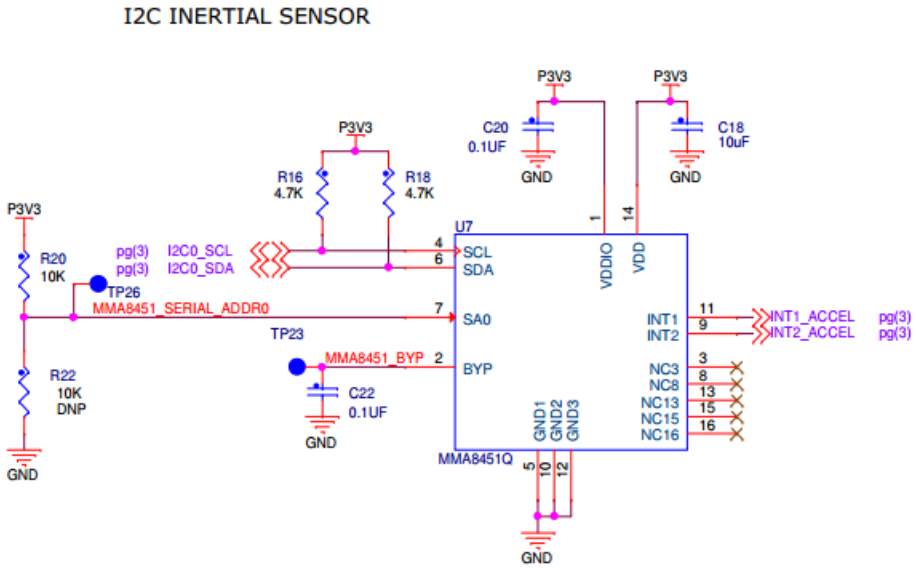
- **Configure o Drive I2C conforme apresentado.**



Acelerômetro

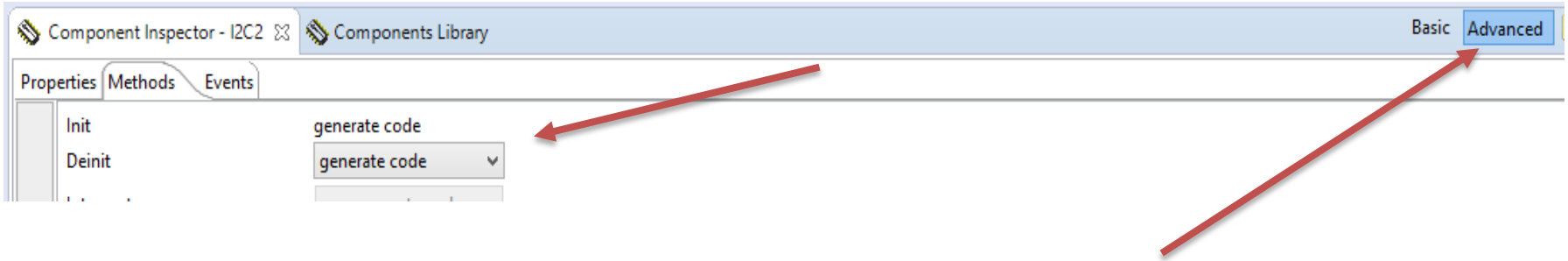
- **Verifique as configurações dos pinos I2C.**





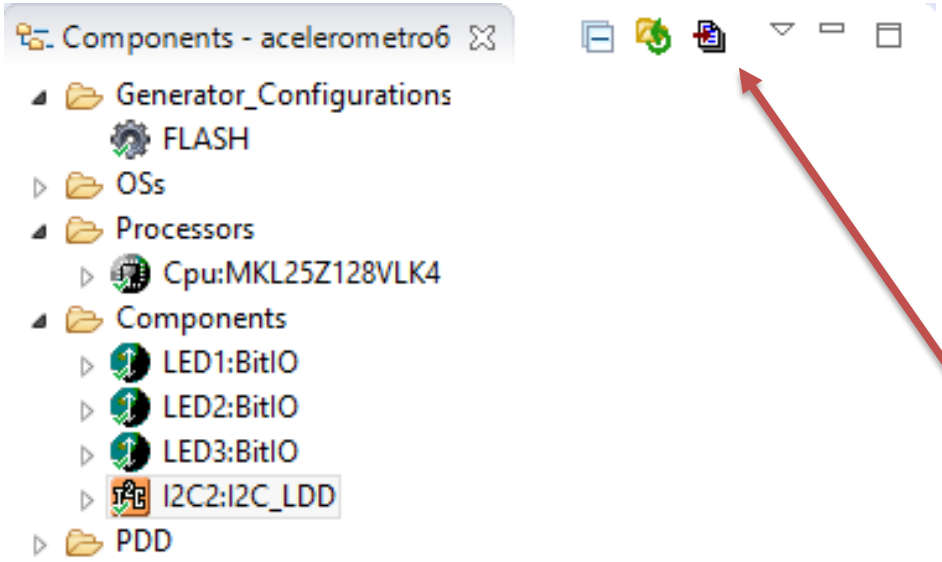
Acelerômetro

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



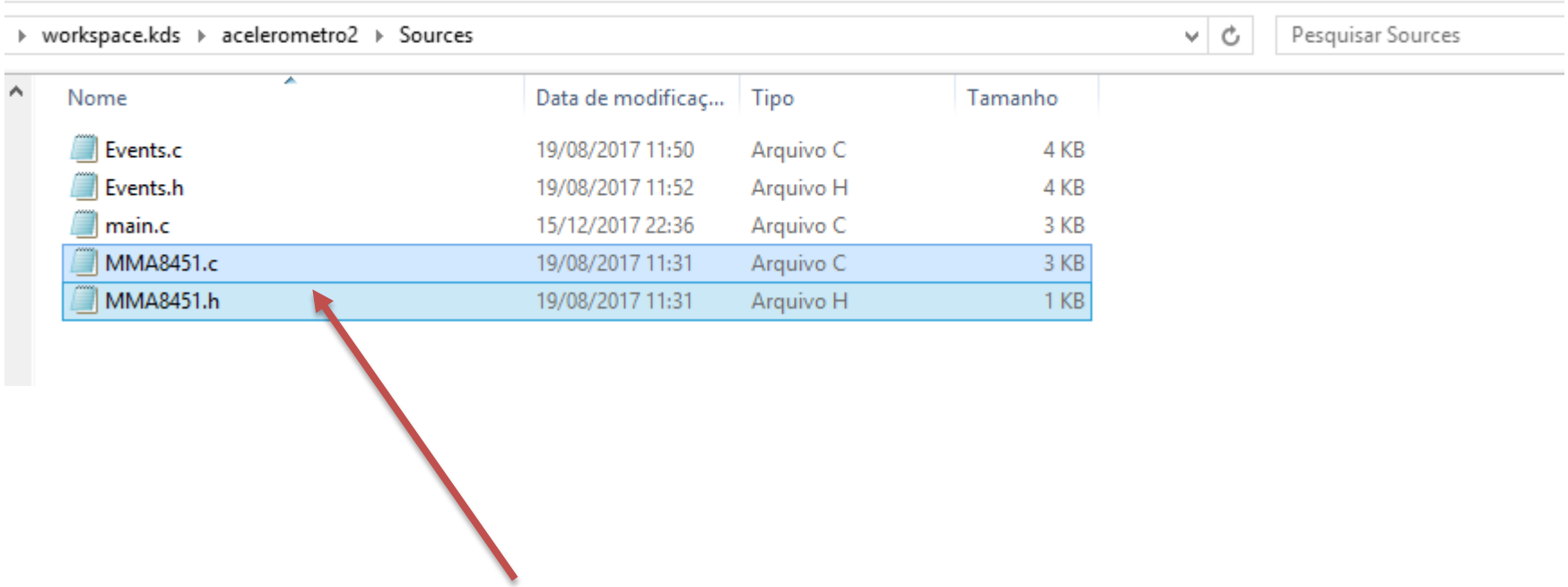
Acelerômetro

- **Clique para gerar os códigos.**



Acelerômetro

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



Acelerômetro

- **Acrescente em “main.c” ...**

MMA8451.c | Events.c | main.c | Events.h | makefile | MMA8451.h | 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” ...**

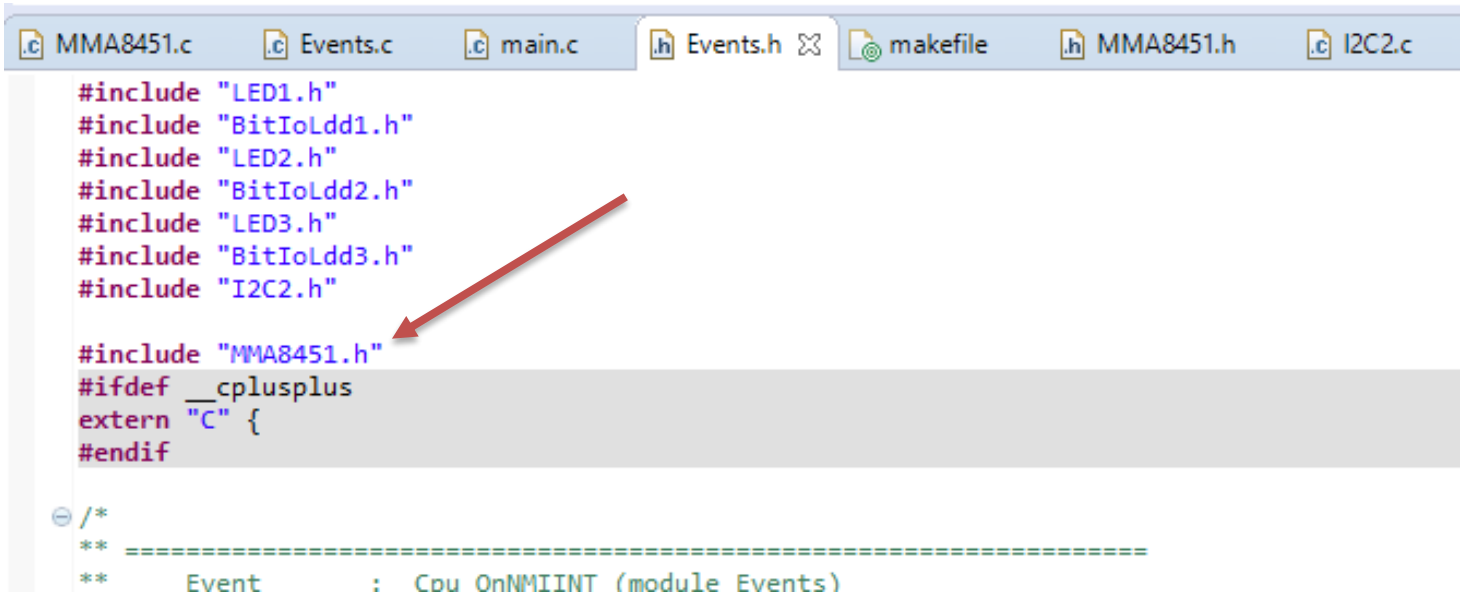
MMA8451.c | Events.c | main.c | Events.h | makefile | MMA8451.h | I2C2.c

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

Acelerômetro

- **Acrescente em “events.h” ...**



Acelerômetro

Compile e Grave e veja os resultados!