Hi, I'm kinda new to MCU's and programming. I'm using STM32CubeMX for configuration and Keilu5 for embedding. I want to communicate with ADXL345 from my STM32F072RBT Nucleoboard via I2C but I'm having some problems.

I have built the circuit properly, Addres pins, pull up resistors, VCC(3V3) and GND is fine. Tied CS! pin to 3V3 for I2C activation. And here is the code:

main.c

```
/**
******************************************************************************
* File Name : main.c
* Description : Main program body
******************************************************************************
** This notice applies to any and all portions of this file
* that are not between comment pairs USER CODE BEGIN and
* USER CODE END. Other portions of this file, whether
* inserted by the user or by software development tools
* are owned by their respective copyright owners.
*
* COPYRIGHT(c) 2017 STMicroelectronics
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
******************************************************************************
*/
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "stm32f0xx_hal.h"

/* USER CODE BEGIN Includes */
```

main.c

```
/* USER CODE END Includes */

/* Private variables ---------------------------------------------------------*/
I2C_HandleTypeDef hi2c1;

/* USER CODE BEGIN PV */

unsigned char RX_BUFFER[1];
uint8_t TX_DATA[2];
unsigned char TEMP_REG;
unsigned char INIT_REG;
unsigned int testInt = 0;
unsigned int testIntI2C = 0;

/* Private variables ---------------------------------------------------------*/

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----------------------------------------------*/

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/*
#define ADXL_ADD 0x53
#define ADXL_ADD_RD 0xA7
#define ADXL_ADD_WR 0xA6
*/

#define ADXL_ADD 0x1D
#define ADXL_ADD_RD 0x3B
#define ADXL_ADD_WR 0x3A
//#define ADXL_ADD_WR 0xAA

/* USER CODE END 0 */

int main(void)
{

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */

/* MCU Configuration--------------------------------------------------------*/

/* Reset of all peripherals, Initializes the Flash interface and the Systick. */
HAL_Init();

/* USER CODE BEGIN Init */

/* USER CODE END Init */
```

main.c

```
/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_I2C1_Init();

/* USER CODE BEGIN 2 */
TX_DATA[0] = 0x00;
TX_DATA[1] = 0xBB;

RX_BUFFER[0] = 0xFF;
INIT_REG = RX_BUFFER[0];
HAL_I2C_Master_Transmit(&hi2c1, ADXL_ADD_WR, TX_DATA, 2, 200);
testInt = 1;
//HAL_Delay(2);
//testInt = 2;
HAL_I2C_Master_Receive(&hi2c1, ADXL_ADD_RD, RX_BUFFER, 2, 200);
testInt = 3;
TEMP_REG = RX_BUFFER[0];
testInt = 4;
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
HAL_Delay(2);
HAL_I2C_Master_Transmit(&hi2c1, ADXL_ADD_WR, TX_DATA, 1, 200);
//HAL_Delay(2);
HAL_I2C_Master_Receive(&hi2c1, ADXL_ADD_RD, RX_BUFFER, 1, 200);
TEMP_REG = RX_BUFFER[0];
HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_8);
//testInt++;
//TEMP_REG = RX_BUFFER[0];


}
/* USER CODE END 3 */

}

/** System Clock Configuration
*/
void SystemClock_Config(void)
{

RCC_OscInitTypeDef RCC_OscInitStruct;
RCC_ClkInitTypeDef RCC_ClkInitStruct;
RCC_PeriphCLKInitTypeDef PeriphClkInit;
```

main.c

```
/**Initializes the CPU, AHB and APB busses clocks
*/
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = 16;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL12;
RCC_OscInitStruct.PLL.PREDIV = RCC_PREDIV_DIV2;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
_Error_Handler(__FILE__, __LINE__);
}

/**Initializes the CPU, AHB and APB busses clocks
*/
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
{
_Error_Handler(__FILE__, __LINE__);
}

PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_I2C1;
PeriphClkInit.I2c1ClockSelection = RCC_I2C1CLKSOURCE_HSI;
if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
{
_Error_Handler(__FILE__, __LINE__);
}

/**Configure the Systick interrupt time
*/
HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

/**Configure the Systick
*/
HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/* I2C1 init function */
static void MX_I2C1_Init(void)
{

hi2c1.Instance = I2C1;
hi2c1.Init.Timing = 0x2000090E;
hi2c1.Init.OwnAddress1 = 0;
hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
hi2c1.Init.OwnAddress2 = 0;
hi2c1.Init.OwnAddress2Masks = I2C_OA2_NOMASK;
```

main.c

```
hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
if (HAL_I2C_Init(&hi2c1) != HAL_OK)
{
_Error_Handler(__FILE__, __LINE__);

}
testIntI2C++;
/**Configure Analogue filter
*/
if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1, I2C_ANALOGFILTER_ENABLE) != HAL_OK)
{
_Error_Handler(__FILE__, __LINE__);
}

/**Configure Digital filter
*/
if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK)
{
_Error_Handler(__FILE__, __LINE__);
}

}

/** Configure pins as
* Analog
* Input
* Output
* EVENT_OUT
* EXTI
*/
static void MX_GPIO_Init(void)
{

GPIO_InitTypeDef GPIO_InitStruct;

/* GPIO Ports Clock Enable */
__HAL_RCC_GPIOF_CLK_ENABLE();
__HAL_RCC_GPIOA_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);

/*Configure GPIO pin : PA8 */
GPIO_InitStruct.Pin = GPIO_PIN_8;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pin : PB5 */
GPIO_InitStruct.Pin = GPIO_PIN_5;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
```

main.c

```
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
* @brief This function is executed in case of error occurrence.
* @param None
* @retval None
*/
void _Error_Handler(char * file, int line)
{
/* USER CODE BEGIN Error_Handler_Debug */
/* User can add his own implementation to report the HAL error return state */
while(1)
{
}
/* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT

/**
* @brief Reports the name of the source file and the source line number
* where the assert_param error has occurred.
* @param file: pointer to the source file name
* @param line: assert_param error line source number
* @retval None
*/
void assert_failed(uint8_t* file, uint32_t line)
{
/* USER CODE BEGIN 6 */
/* User can add his own implementation to report the file name and line number,
ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
/* USER CODE END 6 */

}

#endif

/**
* @}
*/

/**
* @}
*/

/************************ (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

stm32f0xx_it.c

```
/**
******************************************************************************
* @file stm32f0xx_it.c
* @brief Interrupt Service Routines.
******************************************************************************
*
* COPYRIGHT(c) 2017 STMicroelectronics
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
******************************************************************************
*/
/* Includes ------------------------------------------------------------------*/
#include "stm32f0xx_hal.h"
#include "stm32f0xx.h"
#include "stm32f0xx_it.h"

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/* External variables --------------------------------------------------------*/
extern I2C_HandleTypeDef hi2c1;

/******************************************************************************/
/* Cortex-M0 Processor Interruption and Exception Handlers */
/******************************************************************************/

/**
* @brief This function handles System service call via SWI instruction.
*/
void SVC_Handler(void)
```

```
{
/* USER CODE BEGIN SVC_IRQn 0 */

/* USER CODE END SVC_IRQn 0 */
/* USER CODE BEGIN SVC_IRQn 1 */

/* USER CODE END SVC_IRQn 1 */
}

/**
* @brief This function handles Pendable request for system service.
*/
void PendSV_Handler(void)
{
/* USER CODE BEGIN PendSV_IRQn 0 */

/* USER CODE END PendSV_IRQn 0 */
/* USER CODE BEGIN PendSV_IRQn 1 */

/* USER CODE END PendSV_IRQn 1 */
}

/**
* @brief This function handles System tick timer.
*/
void SysTick_Handler(void)
{
/* USER CODE BEGIN SysTick_IRQn 0 */

/* USER CODE END SysTick_IRQn 0 */
HAL_IncTick();
HAL_SYSTICK_IRQHandler();
/* USER CODE BEGIN SysTick_IRQn 1 */

/* USER CODE END SysTick_IRQn 1 */
}

/******************************************************************************/
/* STM32F0xx Peripheral Interrupt Handlers */
/* Add here the Interrupt Handlers for the used peripherals. */
/* For the available peripheral interrupt handler names, */
/* please refer to the startup file (startup_stm32f0xx.s). */
/******************************************************************************/

/**
* @brief This function handles RCC and CRS global interrupts.
*/
void RCC_CRS_IRQHandler(void)
{
/* USER CODE BEGIN RCC_CRS_IRQn 0 */

/* USER CODE END RCC_CRS_IRQn 0 */
/* USER CODE BEGIN RCC_CRS_IRQn 1 */

/* USER CODE END RCC_CRS_IRQn 1 */
}

/**
* @brief This function handles I2C1 event global interrupt / I2C1 wake-up interrupt through EXTI line 23.
```

```
*/
void I2C1_IRQHandler(void)
{
/* USER CODE BEGIN I2C1_IRQn 0 */

/* USER CODE END I2C1_IRQn 0 */
if (hi2c1.Instance->ISR & (I2C_FLAG_BERR | I2C_FLAG_ARLO | I2C_FLAG_OVR)) {
HAL_I2C_ER_IRQHandler(&hi2c1);
} else {
HAL_I2C_EV_IRQHandler(&hi2c1);
}
/* USER CODE BEGIN I2C1_IRQn 1 */

/* USER CODE END I2C1_IRQn 1 */
}

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */
/************************ (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```

stm32f0xx_hal_msp.c

```
/**
******************************************************************************
* File Name : stm32f0xx_hal_msp.c
* Description : This file provides code for the MSP Initialization
* and de-Initialization codes.
******************************************************************************
** This notice applies to any and all portions of this file
* that are not between comment pairs USER CODE BEGIN and
* USER CODE END. Other portions of this file, whether
* inserted by the user or by software development tools
* are owned by their respective copyright owners.
*
* COPYRIGHT(c) 2017 STMicroelectronics
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR
```

stm32f0xx_hal_msp.c

```
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
******************************************************************************
*/
/* Includes ------------------------------------------------------------------*/
#include "stm32f0xx_hal.h"

extern void _Error_Handler(char *, int);
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */
/**
* Initializes the Global MSP.
*/
void HAL_MspInit(void)
{
/* USER CODE BEGIN MspInit 0 */

/* USER CODE END MspInit 0 */

__HAL_RCC_SYSCFG_CLK_ENABLE();

/* System interrupt init*/
/* SVC_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SVC_IRQn, 0, 0);
/* PendSV_IRQn interrupt configuration */
HAL_NVIC_SetPriority(PendSV_IRQn, 0, 0);
/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);

/* Peripheral interrupt init */
/* RCC_CRS_IRQn interrupt configuration */
HAL_NVIC_SetPriority(RCC_CRS_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(RCC_CRS_IRQn);

/* USER CODE BEGIN MspInit 1 */

/* USER CODE END MspInit 1 */
}

void HAL_I2C_MspInit(I2C_HandleTypeDef* hi2c)
{

GPIO_InitTypeDef GPIO_InitStruct;
if(hi2c->Instance==I2C1)
{
/* USER CODE BEGIN I2C1_MspInit 0 */

/* USER CODE END I2C1_MspInit 0 */

/**I2C1 GPIO Configuration
PB8 ------> I2C1_SCL
PB9 ------> I2C1_SDA
```

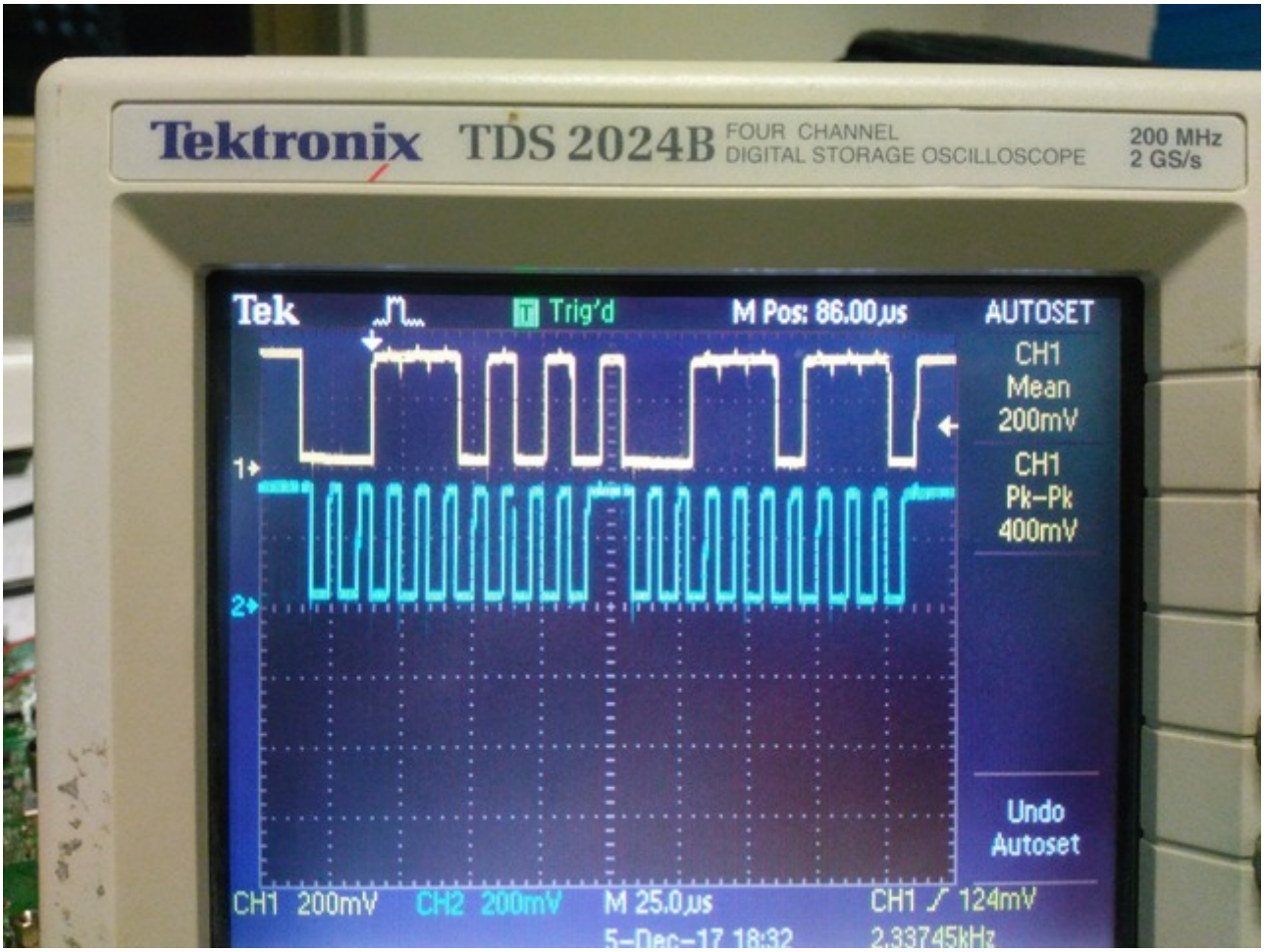
stm32f0xx_hal_msp.c

```
*/
GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9;
GPIO_InitStruct.Mode = GPIO_MODE_AF_OD;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF1_I2C1;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/* Peripheral clock enable */
__HAL_RCC_I2C1_CLK_ENABLE();
/* I2C1 interrupt Init */
HAL_NVIC_SetPriority(I2C1_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(I2C1_IRQn);
/* USER CODE BEGIN I2C1_MspInit 1 */

/* USER CODE END I2C1_MspInit 1 */
}

}

void HAL_I2C_MspDeInit(I2C_HandleTypeDef* hi2c)
{

if(hi2c->Instance==I2C1)
{
/* USER CODE BEGIN I2C1_MspDeInit 0 */

/* USER CODE END I2C1_MspDeInit 0 */
/* Peripheral clock disable */
__HAL_RCC_I2C1_CLK_DISABLE();

/**I2C1 GPIO Configuration
PB8 ------> I2C1_SCL
PB9 ------> I2C1_SDA
*/
HAL_GPIO_DeInit(GPIOB, GPIO_PIN_8|GPIO_PIN_9);

/* I2C1 interrupt DeInit */
HAL_NVIC_DisableIRQ(I2C1_IRQn);
/* USER CODE BEGIN I2C1_MspDeInit 1 */

/* USER CODE END I2C1_MspDeInit 1 */
}

}

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */

/**
* @}
*/

/**
* @}
*/
```

stm32f0xx_hal_msp.c

/********** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/

This is the waveform:



It transmts 3A then 3B WR and READ addresses. It skips up the following data byte in Transmit.

```
HAL_I2C_Master_Transmit(&hi2c1, ADXL_ADD_WR, TX_DATA, 1, 200);
HAL_I2C_Master_Receive(&hi2c1, ADXL_ADD_RD, RX_BUFFER, 1, 200);
```

This is the code it runs but why it sends only Address then moves on? I used two different ADXL345 modules so I'm sure that isn't about the ADXL. This is about my code. Can you help me with this? Thanks.