

**Temperature measurement**

Embedded Systems

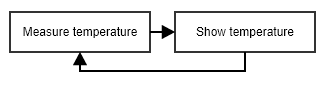
Laboratory work 3

Done by: EKSfmu-16 gr. st. Arūnas Butkus

Checked by: doc. dr. Andrius Ušinskas

**Work goal:** STM3210E-EVAL board measure temperature and show it on the LCD

Algorithms used in tasks.

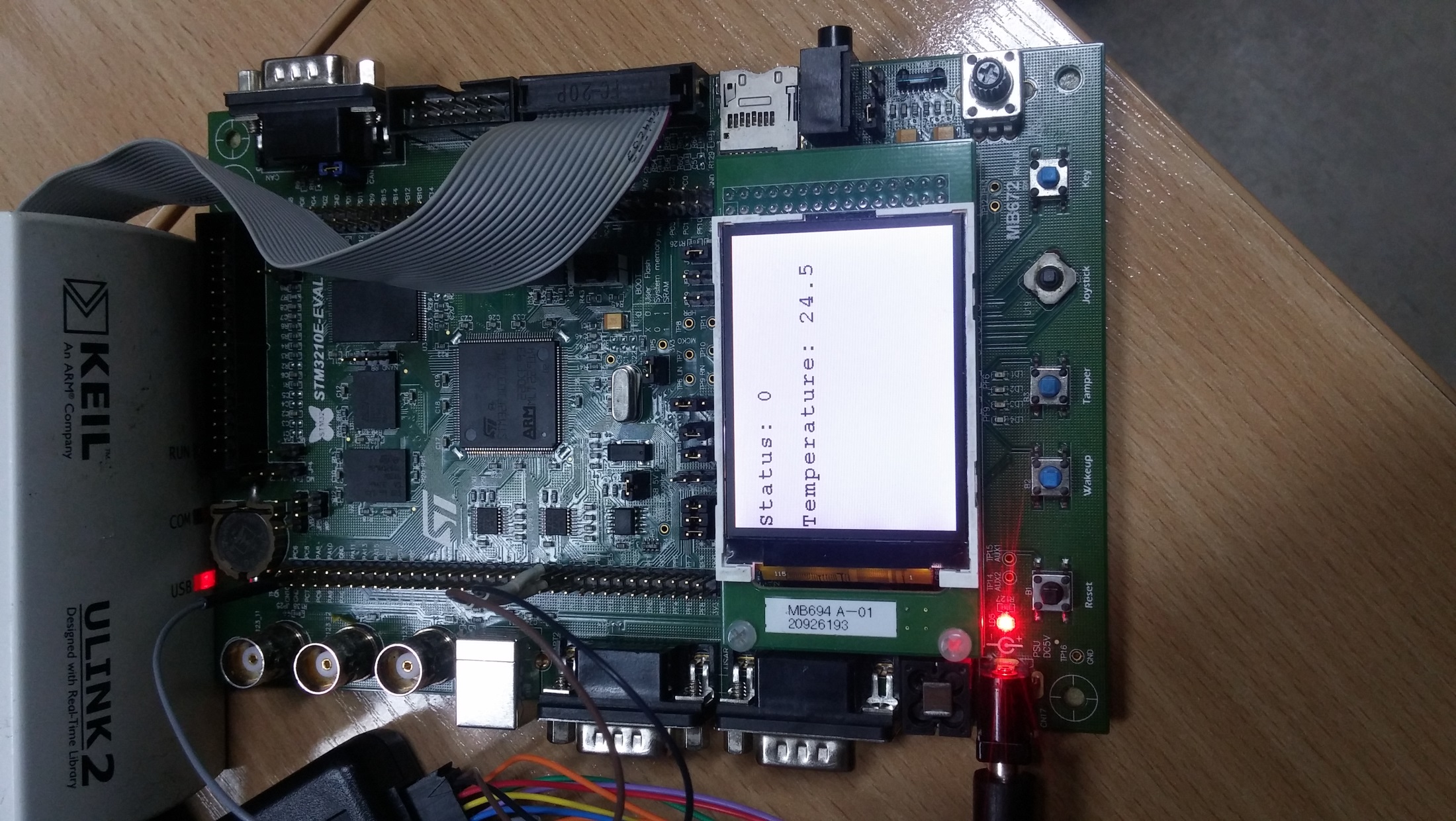


**Fig. 1.** “Algorithm”.

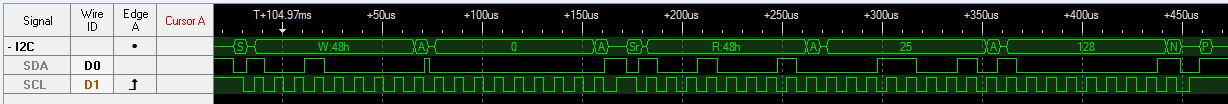
Results analysis.

Temperature comes from LM75 chip over I2C connection – faffed with it a bunch in bachelor as well. Here just using HAL (or whatever) libraries, everything is configured by MX Cube. LCD control libraries are also there from MX Cube. So just passing around variables with minimal processing using functions prepared by other software – no fun in programming, all the annoyance of figuring out how someone else’s code works and making it work in your case.

So there’s results shown on display (Fig. 2.) and message going from LM75 over I2C to the MCU captured with Logic Analyzer (Fig. 3.).



**Fig. 2.** LCD output of temperature measurement.



**Fig. 3.** Logic analyzer output showing LM75 saying that the temperature is 25.5 degrees.

Conclusions.

It works as needed. Meh.

Source code.

001 /\*\*

002 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

003 \* @file Templates/Src/main.c

004 \* @author MCD Application Team

005 \* @version V1.4.0

006 \* @date 29-April-2016

007 \* @brief Main program body

008 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

009 \* @attention

010 \*

011 \* <h2><center>&copy; COPYRIGHT(c) 2016 STMicroelectronics</center></h2>

012 \*

013 \* Redistribution and use in source and binary forms, with or without modification,

014 \* are permitted provided that the following conditions are met:

015 \* 1. Redistributions of source code must retain the above copyright notice,

016 \* this list of conditions and the following disclaimer.

017 \* 2. Redistributions in binary form must reproduce the above copyright notice,

018 \* this list of conditions and the following disclaimer in the documentation

019 \* and/or other materials provided with the distribution.

020 \* 3. Neither the name of STMicroelectronics nor the names of its contributors

021 \* may be used to endorse or promote products derived from this software

022 \* without specific prior written permission.

023 \*

024 \* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

025 \* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

026 \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

027 \* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

028 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

029 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

030 \* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

031 \* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

032 \* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

033 \* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

034 \*

035 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

036 \*/

037

038 /\* Includes ------------------------------------------------------------------\*/

039 #include "main.h"

040 #include "stm3210e\_eval\_lcd.h"

041 #include "stm3210e\_eval\_tsensor.h"

042

043 /\*\* @addtogroup STM32F1xx\_HAL\_Examples

044 \* @{

045 \*/

046

047 /\*\* @addtogroup Templates

048 \* @{

049 \*/

050

051 /\* Private typedef -----------------------------------------------------------\*/

052 /\* Private define ------------------------------------------------------------\*/

053 /\* Private macro -------------------------------------------------------------\*/

054 /\* Private variables ---------------------------------------------------------\*/

055 /\* Private function prototypes -----------------------------------------------\*/

056 void SystemClock\_Config(void);

057 void DisplayString(uint16\_t line, uint8\_t \*pText);

058

059 /\* Private functions ---------------------------------------------------------\*/

060

061 /\*\*

062 \* @brief Main program

063 \* @param None

064 \* @retval None

065 \*/

066 int main(void)

067 {

068

069 /\* STM32F103xG HAL library initialization:

070 - Configure the Flash prefetch

071 - Systick timer is configured by default as source of time base, but user

072 can eventually implement his proper time base source (a general purpose

073 timer for example or other time source), keeping in mind that Time base

074 duration should be kept 1ms since PPP\_TIMEOUT\_VALUEs are defined and

075 handled in milliseconds basis.

076 - Set NVIC Group Priority to 4

077 - Low Level Initialization

078 \*/

079 HAL\_Init();

080

081 /\* Configure the system clock to 72 MHz \*/

082 SystemClock\_Config();

083

084

085 /\* Add your application code here \*/

086

087 \_\_HAL\_RCC\_FSMC\_CLK\_ENABLE();

088 BSP\_LCD\_Init();

089 BSP\_LCD\_Clear(LCD\_COLOR\_WHITE);

090 \_\_HAL\_RCC\_FSMC\_CLK\_DISABLE();

091

092 char buffer[20];

093 uint16\_t temperature;

094 uint8\_t status;

095 uint8\_t point5;

096

097 TSENSOR\_IO\_Init();

098 BSP\_TSENSOR\_Init();

099 status = BSP\_TSENSOR\_ReadStatus();

100 sprintf(buffer,"Status: %d", status);

101 DisplayString(1, buffer);

102

103 /\* Infinite loop \*/

104 while (1)

105 {

106 temperature = BSP\_TSENSOR\_ReadTemp();

107 point5 = temperature % 2;

108 temperature = temperature / 2;

109 if (point5 > 0) point5 = 5;

110 sprintf(buffer,"Temperature: %d.%d", temperature, point5);

111 DisplayString(3, buffer);

112 HAL\_Delay(100);

113 }

114 }

115

116 void DisplayString(uint16\_t line, uint8\_t \*pText) {

117 \_\_HAL\_RCC\_FSMC\_CLK\_ENABLE();

118 BSP\_LCD\_DisplayStringAtLine(line, pText);

119 \_\_HAL\_RCC\_FSMC\_CLK\_DISABLE();

120 }

121

122 /\*\*

123 \* @brief System Clock Configuration

124 \* The system Clock is configured as follow :

125 \* System Clock source = PLL (HSE)

126 \* SYSCLK(Hz) = 72000000

127 \* HCLK(Hz) = 72000000

128 \* AHB Prescaler = 1

129 \* APB1 Prescaler = 2

130 \* APB2 Prescaler = 1

131 \* HSE Frequency(Hz) = 8000000

132 \* HSE PREDIV1 = 1

133 \* PLLMUL = 9

134 \* Flash Latency(WS) = 2

135 \* @param None

136 \* @retval None

137 \*/

138 void SystemClock\_Config(void)

139 {

140 RCC\_ClkInitTypeDef clkinitstruct = {0};

141 RCC\_OscInitTypeDef oscinitstruct = {0};

142

143 /\* Enable HSE Oscillator and activate PLL with HSE as source \*/

144 oscinitstruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSE;

145 oscinitstruct.HSEState = RCC\_HSE\_ON;

146 oscinitstruct.HSEPredivValue = RCC\_HSE\_PREDIV\_DIV1;

147 oscinitstruct.PLL.PLLState = RCC\_PLL\_ON;

148 oscinitstruct.PLL.PLLSource = RCC\_PLLSOURCE\_HSE;

149 oscinitstruct.PLL.PLLMUL = RCC\_PLL\_MUL9;

150 if (HAL\_RCC\_OscConfig(&oscinitstruct)!= HAL\_OK)

151 {

152 /\* Initialization Error \*/

153 while(1);

154 }

155

156 /\* Select PLL as system clock source and configure the HCLK, PCLK1 and PCLK2

157 clocks dividers \*/

158 clkinitstruct.ClockType = (RCC\_CLOCKTYPE\_SYSCLK | RCC\_CLOCKTYPE\_HCLK | RCC\_CLOCKTYPE\_PCLK1 | RCC\_CLOCKTYPE\_PCLK2);

159 clkinitstruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_PLLCLK;

160 clkinitstruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;

161 clkinitstruct.APB2CLKDivider = RCC\_HCLK\_DIV1;

162 clkinitstruct.APB1CLKDivider = RCC\_HCLK\_DIV2;

163 if (HAL\_RCC\_ClockConfig(&clkinitstruct, FLASH\_LATENCY\_2)!= HAL\_OK)

164 {

165 /\* Initialization Error \*/

166 while(1);

167 }

168 }

169

170

171 #ifdef USE\_FULL\_ASSERT

172

173 /\*\*

174 \* @brief Reports the name of the source file and the source line number

175 \* where the assert\_param error has occurred.

176 \* @param file: pointer to the source file name

177 \* @param line: assert\_param error line source number

178 \* @retval None

179 \*/

180 void assert\_failed(uint8\_t\* file, uint32\_t line)

181 {

182 /\* User can add his own implementation to report the file name and line number,

183 ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) \*/

184

185 /\* Infinite loop \*/

186 while (1)

187 {

188 }

189 }

190 #endif

191

192 /\*\*

193 \* @}

194 \*/

195

196 /\*\*

197 \* @}

198 \*/

199

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

201

Electrical schemes:

