

ESD

Lab Assignment 2

Name: Shobhit Kumar

BITS ID: 2021MT12072

Q1. What is the smallest change in input voltage that the ADC can detect?(+Vref=3.3 V)

Answer

Since we have 12-bit ADC in STM32F103RB, the maximum number of input voltages that can be detected is

$2^{12} = 4096$ different values from 0 – 4095.

Since Vref = 3.3 V so the smallest change that can be detected is

$3.3/4096 = 0.8$ milli Volts.

Answer = 0.8mV

Q2. What is the clock frequency of ADC ?

Answer

In the project, I have configured the crystal xTal as 8 MHz

This provides a PCLK2 or APB2 as 72 MHz

While programming the ADC we need to keep ADC Clock less than or equal to 14MHz. So programmed the ADC Pre-scalar divisor in RCC->CFGR |= (2 << 14) which enables a scalar divisor of 6

This gives ADC Clock = 72 MHz / 6 = 12 MHz

Answer = 12 MHz

Q3. Give the steps to program timer for 2 second delay generation with calculation

Answer

$xTAL = 8 \text{ MHz} \Rightarrow PCLK2/APB2 = 72 \text{ MHz}$

So if we put pre-scalar value = 7200 -1 we get

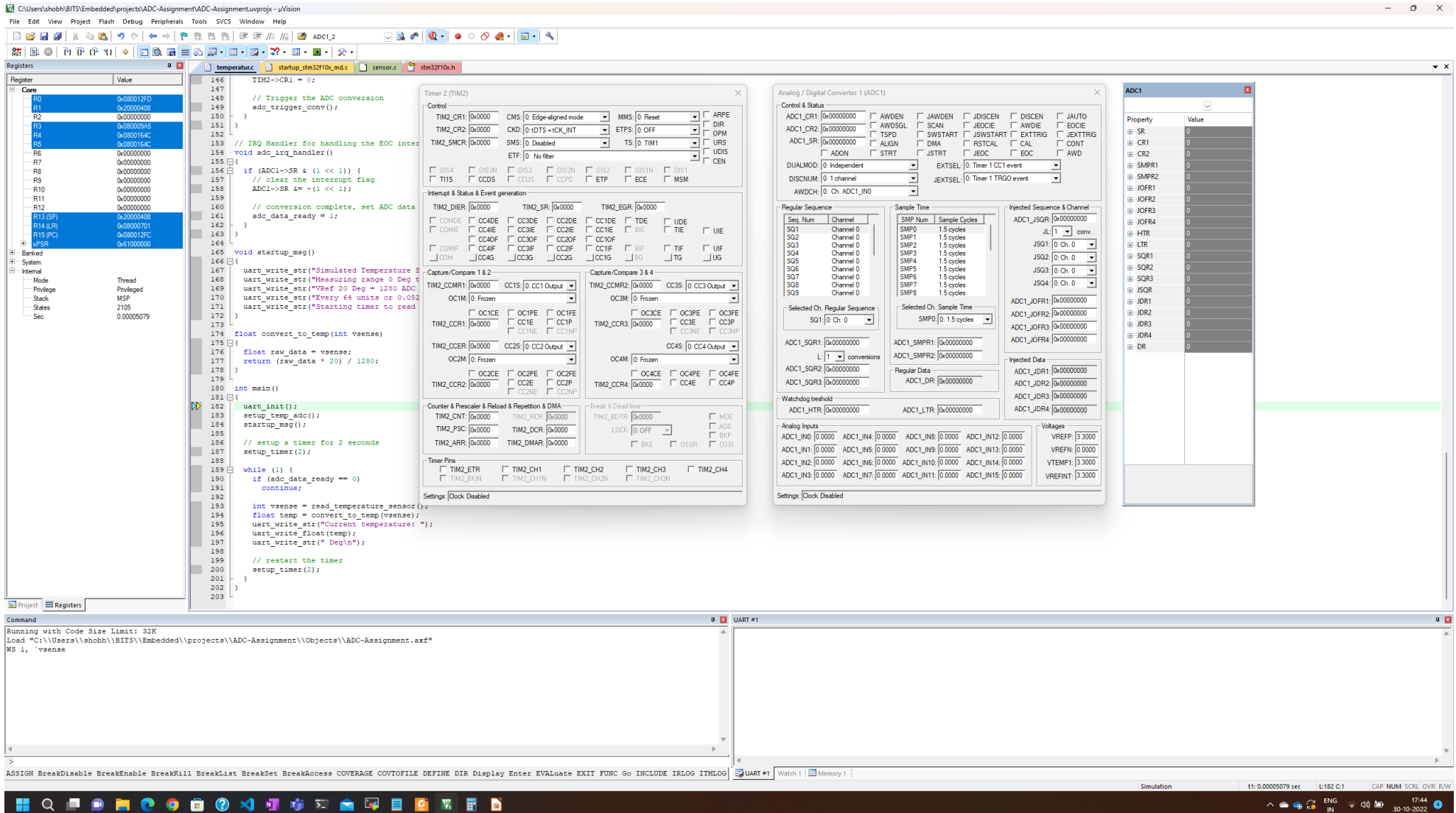
$\text{delay} = (PSC + 1) / 72\text{MHz} = 7200/72000000 = 1/10000\text{s} = 0.1\text{ms}$

Now to get the delay of 2 seconds we need to load ARR as

$ARR + 1 = \text{delay} / PSC = 2000 / 0.1 = 20000$

So $ARR = 20000 - 1 = 19999$

1. Initial Screen before start



2. Sensor characteristics printed on UART

Registers

Register	Value
Core	
R0	0x00000000
R1	0x0013804
R2	0x0001562
R3	0x00000006
R4	0x0000164C
R5	0x0000164C
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000408
R14 (LR)	0x00001621
R15 (PC)	0x00001385
PSR	0x1000000
Banked	
System	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	17181039
Sec	0.23864710

temperature

startups_stm32f10x_md5

sensor.c

stm32f10x.h

```
146 TIM2->CR1 = 0;
147
148 // Trigger the ADC conversion
149 adc_trigger_conv();
150 }
151
152 // IRQ Handler for handling the EOC interrupt
153 void adc_irq_handler()
154 {
155     if (ADC1->SR & (1 << 1)) {
156         // clear the interrupt flag
157         ADC1->SR &= ~(1 << 1);
158     }
159     // conversion complete, set ADC data
160     adc_data_ready = 1;
161 }
162
163 void startup_msg()
164 {
165     uart_write_str("Simulated Temperature Sensor\n");
166     uart_write_str("Measuring range 0 Deg to -64 Deg\n");
167     uart_write_str("VRef 20 Deg = 1280 ADC\n");
168     uart_write_str("Every 64 units or 0.052\n");
169     uart_write_str("Starting timer to read\n");
170 }
171
172 float convert_to_temp(int vensee)
173 {
174     float raw_data = vensee;
175     return (raw_data * 20) / 1280;
176 }
177
178 int main()
179 {
180     uart_init();
181     setup_timer(2);
182     setup_temp_adc();
183     startup_msg();
184     // setup a timer for 2 seconds
185     setup_timer(2);
186     while (1) {
187         if (adc_data_ready == 0)
188             continue;
189         int vensee = read_temperature_sensor();
190         float temp = convert_to_temp(vensee);
191         uart_write_str("Current temperature: ");
192         uart_write_float(temp);
193         uart_write_str(" Deg\n");
194         // restart the timer
195         setup_timer(2);
196     }
197 }
198
199
200
201
202
203
```

Timer 2 (TIM2)

Control

TIM2_CR1: 0x0000 CMS: 0 Edge-aligned mode MMS: 0 Reset ARPE: 0 DIR: 0 OPM: 0 TIM2_CR2: 0x0000 CKD: 0 10TS + CK_INT ETPS: 0 OFF TIM2_SMCR: 0x0000 SMS: 0 Disabled TS: 0 TIM1 UIS: 0 EOC CEN: 0 ETRF: 0 No filter

Interrupt & Status & Event generation

TIM2_DIER: 0x0000 TIM2_SR: 0x0000 TIM2_EGR: 0x0000

Capture/Compare 1 & 2

TIM2_CCMR1: 0x0000 OC1S: 0 CC1 Output TIM2_CR1: 0x0000 OC1CE: 0 OC1PE: 0 OC1FE: 0 OC1NE: 0 OC1NP: 0 TIM2_CCR1: 0x0000 OC2S: 0 CC2 Output OC2CE: 0 OC2PE: 0 OC2FE: 0 OC2NE: 0 OC2NP: 0

Counter & Prescaler & Reload & Repetition & DMA

TIM2_CNT: 0x0000 TIM2_PSC: 0x0000 TIM2_ARR: 0x0000 TIM2_DCR: 0x0000 TIM2_DMAR: 0x0000

Timer Pins

TIM2_ETR: 0 TIM2_CH1: 0 TIM2_CH2: 0 TIM2_CH3: 0 TIM2_CH4: 0 TIM2_BKIN: 0 TIM2_CH1N: 0 TIM2_CH2N: 0 TIM2_CH3N: 0

Settings: Clock Disabled

Analog to Digital Converter 1 (ADC1)

Control & Status

ADC1_CR1: 0x00000020 AWDEN: 0 JAWDEN: 0 JDISCEN: 0 DISCEN: 0 JAUTO: 0 ADC1_CR2: 0x00000001 AWDSGL: 0 SCAN: 0 JE0CIE: 0 JE0CIE: 0 EXTTRIG: 0 JEXTTRIG: 0 ADC1_SR: 0x00000000 TSOP: 0 DMA: 0 RSTCAL: 0 CAL: 0 CONT: 0 ALIGN: 0 JSTRT: 0 JE0C: 0 EOC: 0 AWD: 0 ADON: 0 STRT: 0

DUALMOD: 0 Independent EXTSEL: 0 Timer 1 CC1 event DISCNUM: 0 1 channel JEXTSEL: 0 Timer 1 TRGO event AWDCCH: 0 Ch. ADC1_IN0

Regular Sequence

Seq Num Channel

SQ1 Channel 1 SQ2 Channel 0 SQ3 Channel 0 SQ4 Channel 0 SQ5 Channel 0 SQ6 Channel 0 SQ7 Channel 0 SQ8 Channel 0 SQ9 Channel 0

Sample Time

SMP Num Sample Cycles

SMP0 1.5 cycles SMP1 28.5 cycles SMP2 1.5 cycles SMP3 1.5 cycles SMP4 1.5 cycles SMP5 1.5 cycles SMP6 1.5 cycles SMP7 1.5 cycles SMP8 1.5 cycles

Injected Sequence & Channel

ADC1_ISR: 0x00000000 JIL 1 conv. JSQ1: 0 Ch. 0 JSQ2: 0 Ch. 0 JSQ3: 0 Ch. 0 JSQ4: 0 Ch. 0

Selected Ch. Regular Sequence

ADC1_SQR1: 0x00000000 ADC1_SQR2: 0x00000000 ADC1_SQR3: 0x00000001

Selected Ch. Sample Time

SMP0 0.15 cycles ADC1_SMPR1: 0x00000000 ADC1_SMPR2: 0x00000018

Regular Data

ADC1_DR: 0x00000000

Watchdog threshold

ADC1_HTR: 0x00000000 ADC1_LTR: 0x00000000

Analog Inputs

ADC1_IN0: 0.0000 ADC1_IN1: 0.0000 ADC1_IN2: 0.0000 ADC1_IN3: 0.0000 ADC1_IN4: 0.0000 ADC1_IN5: 0.0000 ADC1_IN6: 0.0000 ADC1_IN7: 0.0000 ADC1_IN8: 0.0000 ADC1_IN9: 0.0000 ADC1_IN10: 0.0000 ADC1_IN11: 0.0000 ADC1_IN12: 0.0000 ADC1_IN13: 0.0000 ADC1_IN14: 0.0000 ADC1_IN15: 0.0000

Voltages

VREFP: 3.3000 VREFN: 0.0000 VTEMP1: 3.3000 VREFINT: 3.3000

Settings: Clock Disabled, ADCLK: 12.00 MHz

ADC1

Property Value

CR1 0x00000020

CR2 0x00000001

SMPR1 0

SMPR2 0

JOFR1 0

JOFR2 0

JOFR3 0

JOFR4 0

HTR 0

LTR 0

SQR1 0

SQR2 0

SQR3 0

SQR4 0

SQR5 0

SQR6 0

SQR7 0

SQR8 0

SQR9 0

CR1 [Bits 31:0] RW (@ 0x0012404) control register 1

Command

Running with Code Size Limit: 32K

Load "C:\Users\shobh\B\Embedded\projects\ADC-Assignment\Objects\ADC-Assignment.axf"

WS 1, vensee

UART #1

Simulated Temperature Sensor characteristics:
Measuring range 0 Deg to -64 Deg
VRef 20 Deg = 1280 ADC Value => 1.031 Volts
Every 64 units or 0.052 Volts increase amounts to 1 Deg
Starting timer to read temperature value every 2s

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE IRLOG ITMLOG

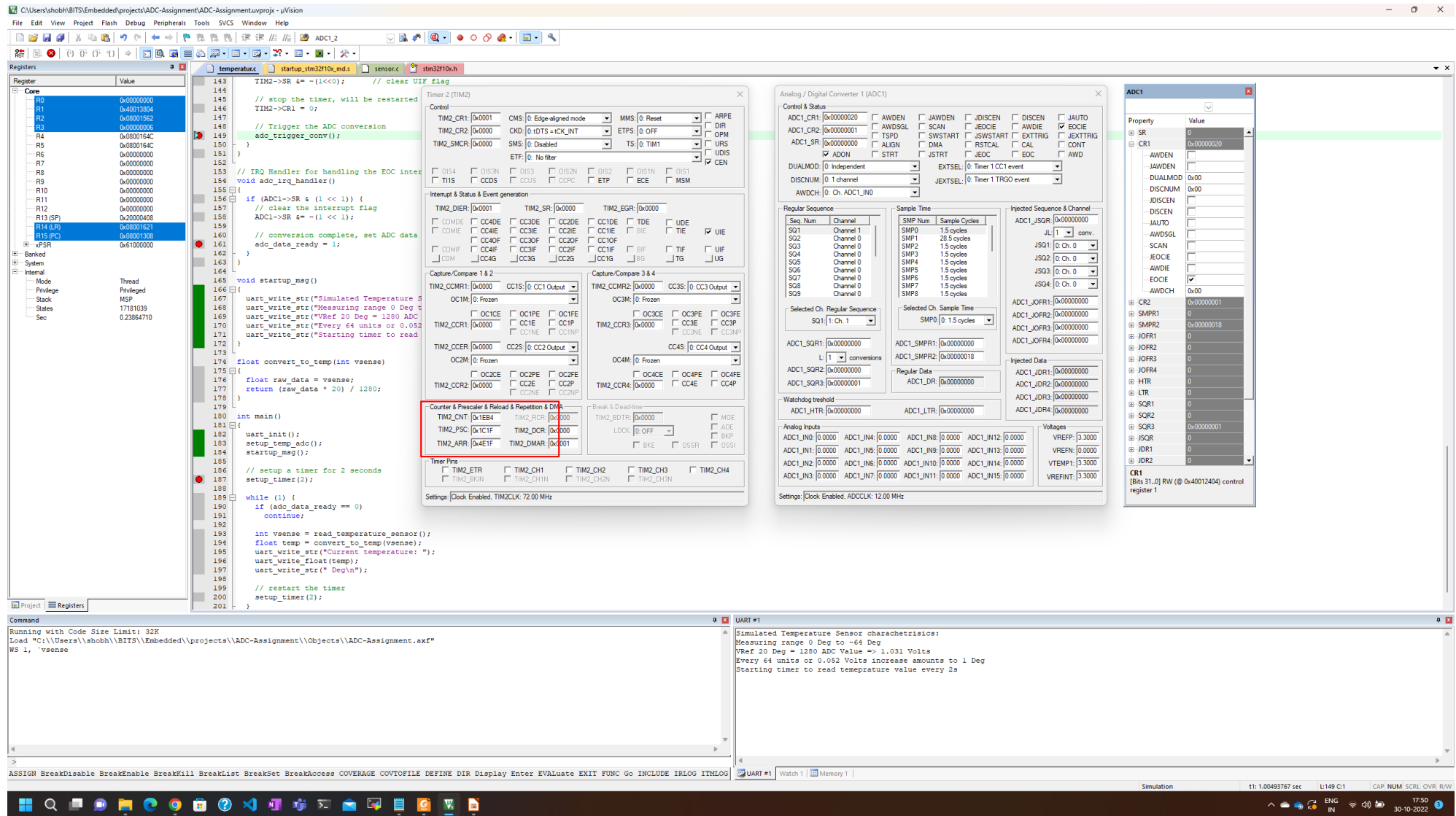
UART #1 Watch 1 Memory 1

Simulation

tt: 0.23864710 sec L187 C1 CAP_NUM SCRL OVBI R/W

17:48 30-10-2022

3. Timer Started



4. Timer Interrupt – Timer expired causing interrupt, trigger ADC conversion

Registers

Register	Value
CORE	0x00000000
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x0000164C
R5	0x0000164C
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00001586
xPSR	0x4100002C

temperature.c

startup_stm3210x_md.s

sensor.c

stm3210x.h

```
130  adc_data_ready = 0;
131  return ADC1->SR;
132
133
134  void adc_trigger_conv()
135  {
136      ADC1->CR2 |= 0x1;
137
138      // Timer2 interrupt handler to timer update
139      void timer_irq_handler()
140      {
141          if (TIM2->SR & 1) {
142              TIM2->SR &= ~(1<<0); // clear UIF
143
144              // stop the timer, will be restarted
145              TIM2->CR1 = 0;
146
147              // Trigger the ADC conversion
148              adc_trigger_conv();
149          }
150      }
151
152      // IRQ Handler for handling the EOC interrupt
153      void adc_irq_handler()
154      {
155          if (ADC1->SR & (1<<1)) {
156              // clear the interrupt flag
157              ADC1->SR &= ~(1<<1);
158
159              // conversion complete, set ADC data
160              adc_data_ready = 1;
161          }
162      }
163
164
165  void startup_msg()
166  {
167      uart_write_str("Simulated Temperature Sensor\n");
168      uart_write_str("Measuring range 0 Deg to 1280 ADC\n");
169      uart_write_str("VRef 20 Deg = 1280 ADC Value => 1.031 Volts\n");
170      uart_write_str("Every 64 units or 0.052 Volts increase amounts to 1 Deg\n");
171      uart_write_str("Starting timer to read temperature value every 2s\n");
172
173
174  float convert_to_temp(int vsense)
175  {
176      float raw_data = vsense;
177      return (raw_data * 20) / 1280;
178
179
180  int main()
181  {
182      uart_init();
183      setup_temp_adc();
184      startup_msg();
185
186      // setup a timer for 2 seconds
187      setup_timer(2);
188  }
```

Timer 2 (TIM2)

Control

Interrupt & Status & Event generation

Capture/Compare 1 & 2

Capture/Compare 3 & 4

Counter & Prescaler & Reload & Repetition & DMA

Timer Pres

Settings

Analog / Digital Converter 1 (ADC1)

Control & Status

Regular Sequence

Sample Time

Injected Sequence & Channel

Selected Ch. Regular Sequence

Selected Ch. Sample Time

ADC1_SMPR1

ADC1_SMPR2

ADC1_DR

Watchdog threshold

Analog Inputs

Voltages

Settings

ADC1

Property

Value

Command

UART #1

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVTOFILE DEFINE DIR Display Enter EVALUATE EXIT FUNC Go INCLUDE IRLOG ITMLOG

UART #1

Watch | Memory |

Simulation

11: 2.2364806 sec

L149 C1

CAP NUM CTRL OVR: RW

ENG IN

17:52

30-10-2022

5. ADC Interrupt – ADC conversion completed. Mark data ready to be read

Registers

Register	Value
R0	0x00000010
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000038
R14 (LR)	0xFFFFFFF9
R15 (PC)	0x00000000
IPSR	0x01000002

temperature.c

startup_stm3210x_md.c

sensor.c

stm3210x.h

```
142 if (TIM2->SR & 1) {
143     TIM2->SR = ~(1<<0); // clear UI
144
145     // stop the timer, will be restarted
146     TIM2->CR1 = 0;
147
148     // Trigger the ADC conversion
149     adc_trigger_conv();
150 }
151
152 // IRQ Handler for handling the EOC interrupt
153 void adc_irq_handler()
154 {
155     if (ADC1->SR & (1<<1)) {
156         // clear the interrupt flag
157         ADC1->SR &= ~(1<<1);
158
159         // conversion complete, set ADC data
160         adc_data_ready = 1;
161     }
162 }
163
164 void startup_msg()
165 {
166     uart_write_str("Simulated Temperature Sensor characteristics:\n");
167     uart_write_str("Measuring range 0 Deg to +64 Deg\n");
168     uart_write_str("VRef 20 Deg = 1280 ADC\n");
169     uart_write_str("Every 64 units or 0.052 Volts increase amounts to 1 Deg\n");
170     uart_write_str("Starting timer to read temperature value every 2s\n");
171 }
172
173 float convert_to_temp(int vsense)
174 {
175     float raw_data = vsense;
176     return (raw_data * 20) / 1280;
177 }
178
179 int main()
180 {
181     uart_init();
182     setup_temp_adc();
183     startup_msg();
184
185     // setup a timer for 2 seconds
186     setup_timer(2);
187
188     while (1) {
189         if (adc_data_ready == 0)
190             continue;
191
192         int vsense = read_temperature_sensor();
193         float temp = convert_to_temp(vsense);
194         uart_write_str("Current temperature: ");
195         uart_write_float(temp);
196         uart_write_str(" Deg\n");
197
198         // restart the timer
199         setup_timer(2);
200     }
```

Control

Timer 2 (TIM2)

Interrupt & Status & Event generation

Capture/Compare 1 & 2

Capture/Compare 3 & 4

Counter & Prescaler & Reload & Repetition & DMA

Time Pres

Settings

Analog / Digital Converter 1 (ADC1)

Control & Status

Regular Sequence

Sample Time

Injected Sequence & Channel

Watchdog threshold

Analog Inputs

Settings

ADC1

Property

Value

Command

Running with Code Size Limit: 32K

Load "C:\Users\shobh\BITS\Embedded\projects\ADC-Assignment\Objects\ADC-Assignment.axf"

WS 1, "vsense"

UART #1

Simulated Temperature Sensor characteristics:

Measuring range 0 Deg to +64 Deg

VRef 20 Deg = 1280 ADC Value => 1.031 Volts

Every 64 units or 0.052 Volts increase amounts to 1 Deg

Starting timer to read temperature value every 2s

Simulation

11: 2.3385100 sec

L161 C1

CAP NUM SCRL OVR: R/W

17:55

30-10-2022

6. UART Prints the temperature reading

Registers

Register	Value
R0	0x00000010
R1	0x0012400
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x000003E8
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00001256
LRSR	0x01000022

temperature.c

```
142 if (TIM2->SR & 1) {
143     TIM2->SR = ~(1<<0); // clear UIF
144 }
145 // stop the timer, will be restarted
146 TIM2->CR1 = 0;
147 // Trigger the ADC conversion
148 adc_trigger_conv();
149 void adc_irq_handler()
150 {
151 }
152 // IRQ Handler for handling the EOC interrupt
153 void adc_irq_handler()
154 {
155     if (ADC1->SR & (1<<1)) {
156         // clear the interrupt flag
157         ADC1->SR = ~(1<<1);
158         // conversion complete, set ADC data
159         adc_data_ready = 1;
160     }
161 }
162 void startup_mag()
163 {
164     uart_write_str("Simulated Temperature Sensor\n");
165     uart_write_str("Measuring range 0 Deg to -64 Deg\n");
166     uart_write_str("Vref 20 Deg = 1280 ADC\n");
167     uart_write_str("Every 64 units or 0.052\n");
168     uart_write_str("Starting timer to read\n");
169 }
170 float convert_to_temp(int vsense)
171 {
172     float raw_data = vsense;
173     return (raw_data * 20) / 1280;
174 }
175 int main()
176 {
177     uart_init();
178     setup_timer(2);
179     startup_mag();
180     // setup a timer for 2 seconds
181     setup_timer(2);
182     while (1) {
183         if (adc_data_ready == 0)
184             continue;
185         int vsense = read_temperature_sensor();
186         float temp = convert_to_temp(vsense);
187         uart_write_str("Current temperature: ");
188         uart_write_float(temp);
189         uart_write_str(" Deg\n");
190         // restart the timer
191         setup_timer(2);
192     }
193 }
```

Registers

Register	Value
R0	0x00000010
R1	0x0012400
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x000003E8
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00001256
LRSR	0x01000022

temperature.c

```
142 if (TIM2->SR & 1) {
143     TIM2->SR = ~(1<<0); // clear UIF
144 }
145 // stop the timer, will be restarted
146 TIM2->CR1 = 0;
147 // Trigger the ADC conversion
148 adc_trigger_conv();
149 void adc_irq_handler()
150 {
151 }
152 // IRQ Handler for handling the EOC interrupt
153 void adc_irq_handler()
154 {
155     if (ADC1->SR & (1<<1)) {
156         // clear the interrupt flag
157         ADC1->SR = ~(1<<1);
158         // conversion complete, set ADC data
159         adc_data_ready = 1;
160     }
161 }
162 void startup_mag()
163 {
164     uart_write_str("Simulated Temperature Sensor\n");
165     uart_write_str("Measuring range 0 Deg to -64 Deg\n");
166     uart_write_str("Vref 20 Deg = 1280 ADC\n");
167     uart_write_str("Every 64 units or 0.052\n");
168     uart_write_str("Starting timer to read\n");
169 }
170 float convert_to_temp(int vsense)
171 {
172     float raw_data = vsense;
173     return (raw_data * 20) / 1280;
174 }
175 int main()
176 {
177     uart_init();
178     setup_timer(2);
179     startup_mag();
180     // setup a timer for 2 seconds
181     setup_timer(2);
182     while (1) {
183         if (adc_data_ready == 0)
184             continue;
185         int vsense = read_temperature_sensor();
186         float temp = convert_to_temp(vsense);
187         uart_write_str("Current temperature: ");
188         uart_write_float(temp);
189         uart_write_str(" Deg\n");
190         // restart the timer
191         setup_timer(2);
192     }
193 }
```

UART #1

```
Simulated Temperature Sensor characteristics:
Measuring range 0 Deg to -64 Deg
Vref 20 Deg = 1280 ADC Value => 1.031 Volts
Every 64 units or 0.052 Volts increase amounts to 1 Deg
Starting timer to read temperature value every 2s
Current temperature: 0.00 Deg
```

ADC1

Property	Value
SR	0x00000010
CR1	0x00000020
AWDEN	0x00
JAWDEN	0x00
DUALMOD	0x00
DISCNUM	0x00
JOSCN	0x00
DISCEN	0x00
JAUTO	0x00
AWDSCAL	0x00
AWDIE	0x00
AWDIF	0x00
AWDCH	0x00
CR2	0x00000001
SMRPR1	0x00000000
SMRPR2	0x00000000
JOFPR1	0x00000000
JOFPR2	0x00000000
JOFPR3	0x00000000
JOFPR4	0x00000000
HTR	0x00000000
LTR	0x00000000
SOFR1	0x00000000
SOFR2	0x00000000
SOFR3	0x00000000
JSOR	0x00000000
JDR1	0x00000000
JDR2	0x00000000
CR1	[Bits 31:0] RW (0x0012404) control register 1

Control & Status

Control & Status	Value
ADCR1	0x00000020
ADCR2	0x00000001
ADCR3	0x00000000
ADCR4	0x00000000
ADCR5	0x00000000
ADCR6	0x00000000
ADCR7	0x00000000
ADCR8	0x00000000
ADCR9	0x00000000
ADCR10	0x00000000
ADCR11	0x00000000
ADCR12	0x00000000
ADCR13	0x00000000
ADCR14	0x00000000
ADCR15	0x00000000
ADCR16	0x00000000
ADCR17	0x00000000
ADCR18	0x00000000
ADCR19	0x00000000
ADCR20	0x00000000
ADCR21	0x00000000
ADCR22	0x00000000
ADCR23	0x00000000
ADCR24	0x00000000
ADCR25	0x00000000
ADCR26	0x00000000
ADCR27	0x00000000
ADCR28	0x00000000
ADCR29	0x00000000
ADCR30	0x00000000
ADCR31	0x00000000
ADCR32	0x00000000
ADCR33	0x00000000
ADCR34	0x00000000
ADCR35	0x00000000
ADCR36	0x00000000
ADCR37	0x00000000
ADCR38	0x00000000
ADCR39	0x00000000
ADCR40	0x00000000
ADCR41	0x00000000
ADCR42	0x00000000
ADCR43	0x00000000
ADCR44	0x00000000
ADCR45	0x00000000
ADCR46	0x00000000
ADCR47	0x00000000
ADCR48	0x00000000
ADCR49	0x00000000
ADCR50	0x00000000
ADCR51	0x00000000
ADCR52	0x00000000
ADCR53	0x00000000
ADCR54	0x00000000
ADCR55	0x00000000
ADCR56	0x00000000
ADCR57	0x00000000
ADCR58	0x00000000
ADCR59	0x00000000
ADCR60	0x00000000
ADCR61	0x00000000
ADCR62	0x00000000
ADCR63	0x00000000
ADCR64	0x00000000
ADCR65	0x00000000
ADCR66	0x00000000
ADCR67	0x00000000
ADCR68	0x00000000
ADCR69	0x00000000
ADCR70	0x00000000
ADCR71	0x00000000
ADCR72	0x00000000
ADCR73	0x00000000
ADCR74	0x00000000
ADCR75	0x00000000
ADCR76	0x00000000
ADCR77	0x00000000
ADCR78	0x00000000
ADCR79	0x00000000
ADCR80	0x00000000
ADCR81	0x00000000
ADCR82	0x00000000
ADCR83	0x00000000
ADCR84	0x00000000
ADCR85	0x00000000
ADCR86	0x00000000
ADCR87	0x00000000
ADCR88	0x00000000
ADCR89	0x00000000
ADCR90	0x00000000
ADCR91	0x00000000
ADCR92	0x00000000
ADCR93	0x00000000
ADCR94	0x00000000
ADCR95	0x00000000
ADCR96	0x00000000
ADCR97	0x00000000
ADCR98	0x00000000
ADCR99	0x00000000
ADCR100	0x00000000
ADCR101	0x00000000
ADCR102	0x00000000
ADCR103	0x00000000
ADCR104	0x00000000
ADCR105	0x00000000
ADCR106	0x00000000
ADCR107	0x00000000
ADCR108	0x00000000
ADCR109	0x00000000
ADCR110	0x00000000
ADCR111	0x00000000
ADCR112	0x00000000
ADCR113	0x00000000
ADCR114	0x00000000
ADCR115	0x00000000
ADCR116	0x00000000
ADCR117	0x00000000
ADCR118	0x00000000
ADCR119	0x00000000
ADCR120	0x00000000
ADCR121	0x00000000
ADCR122	0x00000000
ADCR123	0x00000000
ADCR124	0x00000000
ADCR125	0x00000000
ADCR126	0x00000000
ADCR127	0x00000000
ADCR128	0x00000000
ADCR129	0x00000000
ADCR130	0x00000000
ADCR131	0x00000000
ADCR132	0x00000000
ADCR133	0x00000000
ADCR134	0x00000000
ADCR135	0x00000000
ADCR136	0x00000000
ADCR137	0x00000000
ADCR138	0x00000000
ADCR139	0x00000000
ADCR140	0x00000000
ADCR141	0x00000000
ADCR142	0x00000000
ADCR143	0x00000000
ADCR144	0x00000000
ADCR145	0x00000000
ADCR146	0x00000000
ADCR147	0x00000000
ADCR148	0x00000000
ADCR149	0x00000000
ADCR150	0x00000000
ADCR151	0x00000000
ADCR152	0x00000000
ADCR153	0x00000000
ADCR154	0x00000000
ADCR155	0x00000000
ADCR156	0x00000000
ADCR157	0x00000000
ADCR158	0x00000000
ADCR159	0x00000000
ADCR160	0x00000000
ADCR161	0x00000000
ADCR162	0x00000000
ADCR163	0x00000000
ADCR164	0x00000000
ADCR165	0x00000000
ADCR166	0x00000000
ADCR167	0x00000000
ADCR168	0x00000000
ADCR169	0x00000000
ADCR170	0x00000000
ADCR171	0x00000000
ADCR172	0x00000000
ADCR173	0x00000000
ADCR174	0x00000000
ADCR175	0x00000000
ADCR176	0x00000000
ADCR177	0x00000000
ADCR178	0x00000000
ADCR179	0x00000000
ADCR180	0x00000000
ADCR181	0x00000000
ADCR182	0x00000000
ADCR183	0x00000000
ADCR184	0x00000000
ADCR185	0x00000000
ADCR186	0x00000000
ADCR187	0x00000000
ADCR188	0x00000000
ADCR189	0x00000000
ADCR190	0x00000000
ADCR191	0x00000000
ADCR192	0x00000000
ADCR193	0x00000000
ADCR194	0x00000000
ADCR195	0x00000000
ADCR196	0x00000000
ADCR197	0x00000000
ADCR198	0x00000000
ADCR199	0x00000000
ADCR200	0x00000000
ADCR201	0x00000000
ADCR202	0x00000000
ADCR203	0x00000000
ADCR204	0x00000000
ADCR205	0x00000000
ADCR206	0x00000000
ADCR207	0x00000000
ADCR208	0x00000000
ADCR209	0x00000000
ADCR210	0x00000000
ADCR211	0x00000000
ADCR212	0x00000000
ADCR213	0x00000000
ADCR214	0x00000000
ADCR215	0x00000000
ADCR216	0x00000000
ADCR217	0x00000000
ADCR218	0x00000000
ADCR219	0x00000000
ADCR220	0x00000000
ADCR221	0x00000000
ADCR222	0x00000000
ADCR223	0x00000000
ADCR224	0x00000000
ADCR225	0x00000000
ADCR226	0x00000000
ADCR227	0x00000000
ADCR228	0x00000000
ADCR229	0x00000000
ADCR230	0x00000000
ADCR231	0x00000000
ADCR232	0x00000000
ADCR233	0x00000000
ADCR234	0x00000000
ADCR235	0x00000000
ADCR236	0x00000000
ADCR237	0x00000000
ADCR238	0x00000000
ADCR239	0x00000000
ADCR240	0x00000000
ADCR241	0x00000000
ADCR242	0x00000000
ADCR243	0x00000000
ADCR244	0x00000000
ADCR245	0x00000000
ADCR246	0x00000000
ADCR247	0x00000000
ADCR248	0x00000000
ADCR249	0x00000000
ADCR250	0x00000000
ADCR251	0x00000000
ADCR252	0x00000000
ADCR253	0x00000000
ADCR254	0x00000000
ADCR255	0x00000000
ADCR256	0x00000000
ADCR257	0x00000000
ADCR258	0x00000000
ADCR259	0x00000000
ADCR260	0x00000000
ADCR261	0x00000000
ADCR262	0x00000000
ADCR263	0x00000000
ADCR264	0x00000000
ADCR265	0x00000000
ADCR266	0x00000000
ADCR267	0x00000000
ADCR268	0x00000000
ADCR269	0x00000000
ADCR270	0x00000000
ADCR271	0x00000000
ADCR272	0x00000000
ADCR273	0x00000000
ADCR274	0x00000000
ADCR275	0x00000000
ADCR276	0x00000000
ADCR277	0x00000000
ADCR278	0x00000000
ADCR279	0x00000000
ADCR280	0x00000000
ADCR281	0x00000000
ADCR282	0x00000000
ADCR283	0x00000000
ADCR284	0x00000000
ADCR285	0x00000000
ADCR286	0x00000000
ADCR287	0x00000000
ADCR288	0x00000000
ADCR289	0x00000000
ADCR290	0x00000000
ADCR291	0x00000000
ADCR292	0x00000000
ADCR293	0x00000000
ADCR294	0x00000000
ADCR295	0x00000000
ADCR296	0x00000000
ADCR297	0x00000000
ADCR298	0x00000000
ADCR299	0x00000000
ADCR300	0x00000000
ADCR301	0x00000000
ADCR302	0x00000000
ADCR303	0x00000000
ADCR304	0x00000000
ADCR305	0x00000000
ADCR306	0x00000000
ADCR307	0x00000000
ADCR308	0x00000000
ADCR309	0x00000000
ADCR310	0x00000000
ADCR311	0x00000000
ADCR312	0x00000000
ADCR313	0x00000000
ADCR314	0x00000000
ADCR315	0x00000000
ADCR316	0x00000000
ADCR317	0x00000000
ADCR318	0x00000000
ADCR319	0x00000000
ADCR320	0x00000000
ADCR321	0x00000000
ADCR322	0x00000000
ADCR323	0x00000000
ADCR324	0x00000

7. Change ADC1 IN1 value to 2.3 Volts, Read the new temperature value

The screenshot displays the Keil uVision IDE interface for an STM32F10x microcontroller project. The main window shows the C source code for the ADC1 configuration and simulation. The code includes comments and function calls for setting up the ADC1, including the conversion of raw data to temperature.

On the right side, the 'Registers' window is open, showing the 'ADC1' register set. The 'CR1' register is highlighted, showing its value as 0x00000010. The 'CR2' register is also visible, showing its value as 0x00000001.

At the bottom, the 'Command' window shows the output of the simulation, including the 'Simulated Temperature Sensor characteristics' and the 'Current temperature' value of 44.59 Deg.

8. Some more temperature readings

