GEBZE TECHNICAL UNİVERSİTY

ELECTRONİC ENGINEERING

ELEC334

MICROPROCESSOR

PROJECT 02

PREPARED BY
ÖMER CEBECİ 171024007

1.INTRODUCTION:

In this Project, the code is written for calculator. User can use Keypad for calculation operations and can see results and numbers from SSD. Calculator can do these operations:

*It can add two numbers.

*It can substract two numbers.

*It can multiplication two numbers.

*It can division two numbers.

*It keep in the memory previous result and then it can use this result.

*User can use nagative numbers.(Calculator can operation with negative numbers)

*User also can see nagative floating numbers(-1.24),positive floating numbers(11.24), positive and negative integer numbers (2,-2)

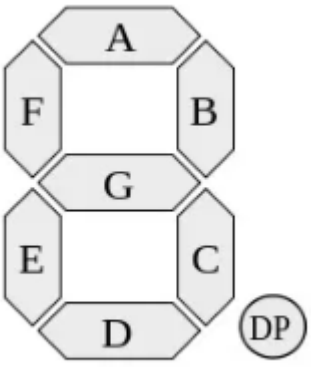
*There are some overflow conditions. For example if result is bigger than 9999 or less than -999 ,user can see '----'.Another overflow condition is scientific overflows.For example , user want to do this operation 'ln(x)',if x<0 ,user will see '----' at SSD.

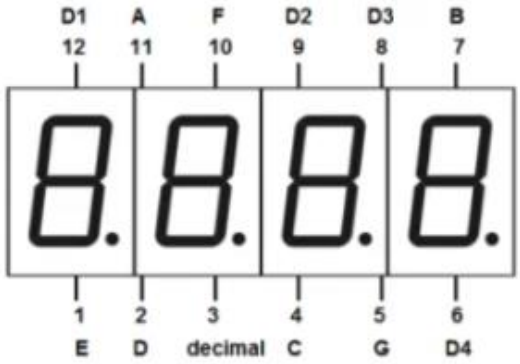
*Calculator have also some initial properties. When board power up, ssd will display '1707',if user don't push any buttons , after ten seconds ssd will close the lights.Or if user push one or two button but don't push '=',ssd will turn back to the idle state that display '1707'.

2.PROCESS OF PROJECT:

2.1 Hardware Setup:

There are two hardware equipments. SSD and Keypad. For SSD:





Board and SSD's connections were done according to the these pictures.

Board and SSD’s pin connections are like that:

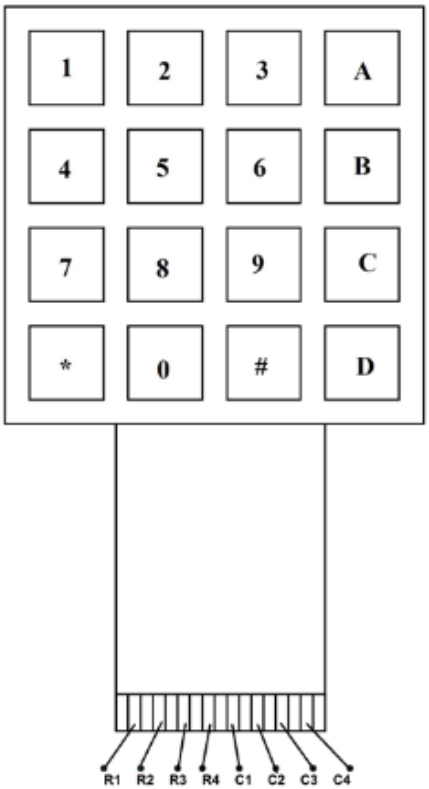
PA0->D1
PA1->D2
PA4->D3
PA5->D4
PA11->DecimalPoint

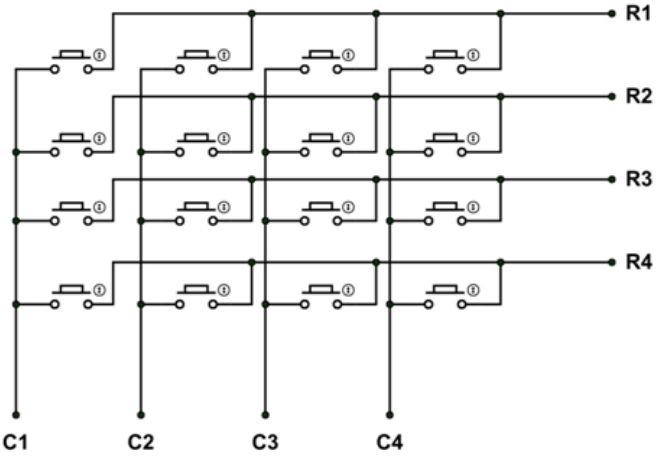
PB0->A
PB1->B
PB2->C
PB3->D
PB4->E
PB5->F
PB6->G

All digits can’t work simultaneous in SSD. To solve this problem code run SSD rapidly so user can see as if working simultaneously.

SSD’s pins and Keypad’s pins are different from each other. Because SSD and Keypad should work simultaneous so they shouldn’t affect each other.

Board and Keypad’s connections were done according to the these pictures.





Keypad’s working principle is like that:

When user push the any button, one of the C1,C2,C3 and C4 will be 1. So code can understand user push from which column . But code have to know user push from which row. To determine the button type(1,2,3,4,5..) .For this reason when user push the button,code check all rows one by one.(only R1 5V others 0V,then R2 only have 5V ,others have 0V….).

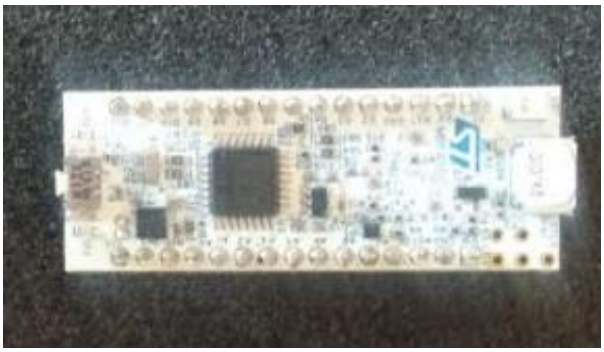
Board and Keypad’s pin connections are like that:

Input Ports:
C1->PB7
C2->PB8
C3->PB9
C4->PA15

Output Ports:
R1->PA6
R2->PA7
R3->PA9
R4->PA10

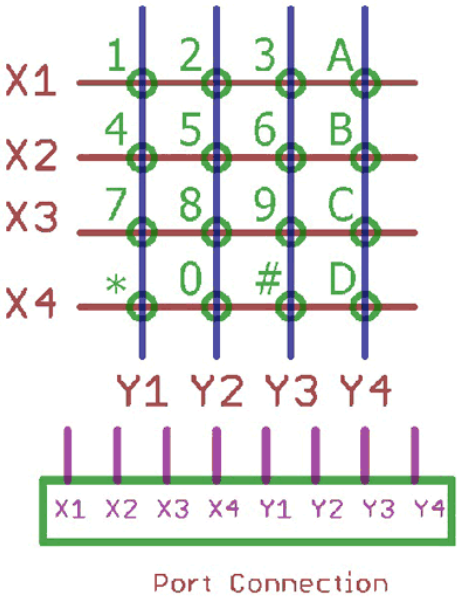
Hardware Picture’s:

4 Digit Seven Segment Display(SSD)



STM32G031K8

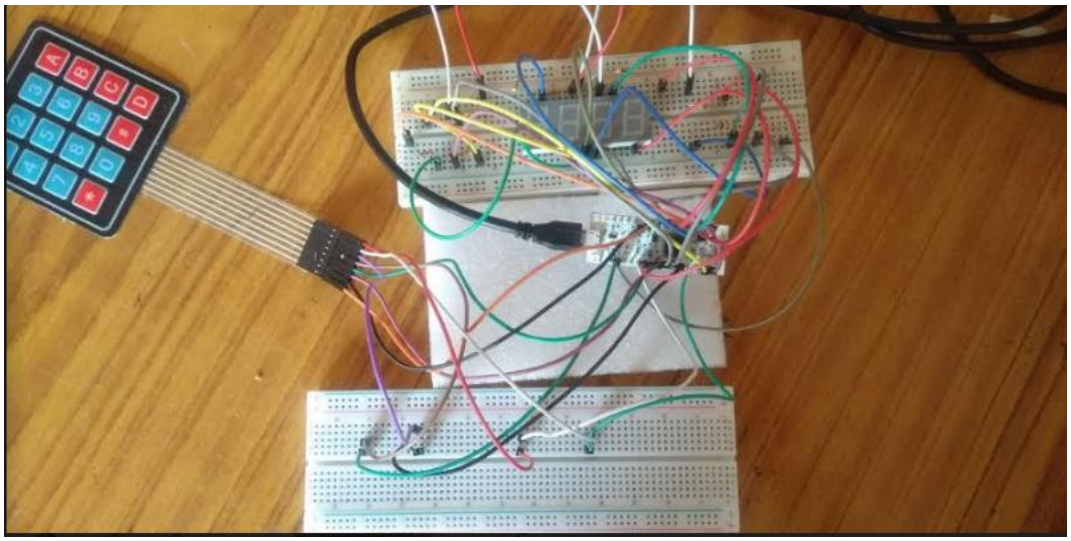
4X4 Keypad



4x4 Keypad PortConnection

,

Man to Man Jumper/Man to Woman Jumper



Full Hardware

2.2 Software Setup:

Below are the project stages.

1)The code written for basic operations (+,-,*,/).A struct was created for stored the numbers. User can only write two number in one operation. After user push the equal button (‘#’),process_result function will work. This function determine of the operation type’s place in array. If operation type is basic operations, first and second number was created and done this operation. There is a code for fisrt number:

```
for(i =0;i<operation_place;i++){
            number_number[0]=number_number[0]+pow(10,i)*number.array_number[operation_place-i-1];
}
```

İ=0:

Number_number[0]=0+10^0.array[operation_place-0-1] this is first digit of first number // ones digit

İ=1

Number_number[0]=0+10^1.array[operation_place-1-1] this is second digit of first number // tens digit

This loop continue until operation_place because after operation place ,there is a second number.

2) The code written for scientific operations(sin cos ln log tan cot sqrt x^2).Same steps with basic operations until process_result function. There is a if-else statement in the function for scientific operations. If operation_type is equal to (*) ,scientific_flag set and pc go to the scientific_process_function. the second number doesn't matter. Operation type was determined in the scientific_process_function. For this reason, There is a loop and every time code checked array if there is an (EA),(EB),(EC),(ED),(EEA),(EEB),(EEC),(EED).

Below There are codes:

```
if(operation_type==15){// For scientific calculations. if operations type is equla 15,user push the button 'E'
    Scientific_Trigonometric_flag=1;

}
```

This code is for determine the scientific operation.

```
for( i =0;i<10;i++){
    if(number.array_number[i]==15&&number.array_number[i+1]==11){// EA for log
        operation_type= 26 ; // 15+11
        break ;
    }
    else if (number.array_number[i]==15 && number.array_number[i+1]==12){ // EB for ln
        operation_type= 27 ; // 15+12
        break ;
    }

    else if (number.array_number[i]==15 && number.array_number[i+1]==13){// EC for sqrt
        operation_type= 28 ; // 15+13
        break ;
    }
    else if (number.array_number[i]==15 && number.array_number[i+1]==14){// ED for x^2
                operation_type= 29 ; // 15+14
                break ;
            }
    else if (number.array_number[i]==15&&number.array_number[i+1]==15){ // trigonometric mode
        if(number.array_number[i+2]==11){ // sinus
            operation_type= 41 ; // 15+15+11
            break ;
        }
        else if(number.array_number[i+2]==12){// cosinus
            operation_type= 42 ; // 15+15+12
            break ;
        }
        else if(number.array_number[i+2]==13){ //tan
            operation_type= 43 ; // 15+15+13
            break ;
        }
        else if (number.array_number[i+2]==14){ // cot
            operation_type= 44 ; // 15+15+14
            break ;
        }
```

This code is for determine the scientific operation's type

3)Floating and Negative Number:

After every operations code checked the result, and if result is not integer Floating_display functions is used.

Understanding floating Numbers:

Result is converted to the integer number and this integer number substract from itself (result)

Below there are codes:

```
tantitive_result=(int)result_old;// tantitive result for find the floating number...
```

```
if(result_old-tantitive_result==0){ // number is normal number there is no floating point
                if(result_old>0){   // if result is positive
                                    Delay(50);
                                    Number_Find(result_old);
                                    Delay(50);
                }
```

If user want to use negative number, first of all it push the 'E' and number. Code check this situation, if array's first number is 15 , that's means is negative number will come.

4)Finally, if user finish an operation and push again (A,B,C,D,E),code determine previous result.

as the first number.

Notes:

If user push A -> in code,11 will come to number.array_number .

If user push B -> in code,12 will come to number.array_number .

If user push C -> in code,13 will come to number.array_number .

If user push D -> in code,14 will come to number.array_number .

If user push *(E) -> in code,15 will come to number.array_number .

If user push # -> in code, PC go to the Process_result function for calculation.

Some Examples:

Basic Operations Examples:

1	2	11	1	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

11->A , 12+111=123

1	2	12	1	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12->B ,12-111=99

1	2	13	1	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

11->C , 12*111=1332

1	2	14	1	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

11->D , 12/111=0.108

Scientific Operations Examples:

1	2	15	11	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (EA)… =LOG(12)

1	2	15	12	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (EB)… =Ln(12)

1	2	15	13	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (EC)… =$\sqrt{12}$

1	2	15	14	1	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (ED)… =(12)*12

1	2	15	15	11	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (EEA)… =sin(12*pi/180)

1	2	15	15	12	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (EEB)… =cos(12*pi/180)

1	2	15	15	13	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (EEC)… =tan(12*pi/180)

1	2	15	15	14	1	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]

12 (EED)… =cot(12*pi/180)

Array[4] and array[5]'values dont matter for all Scientific Operation's(in only first operation)

Two or More Operations Exmples:

1	2	15	11	1	1	#	15	12	0	0	0	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]	array[0]	array[1]	array[2]	array[3]	array[4]	array[6]
		First Operation							Second Operation			

Ln(LOG(12))

1	2	15	11	1	1	#	15	15	13	0	0	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]	array[0]	array[1]	array[2]	array[3]	array[4]	array[6]
		First Operation							Second Operation			

tan(LOG(12)*)

1	2	15	11	1	1	#	15	13	0	0	0	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]	array[0]	array[1]	array[2]	array[3]	array[4]	array[6]
		First Operation							Second Operation			

$$\sqrt{LOG(12)}$$

1	2	15	11	1	1	#	14	0	0	0	0	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]	array[6]	array[0]	array[1]	array[2]	array[3]	array[4]	array[6]
		First Operation							Second Operation			

LOG(12)/0 = '----' (OVERFLOW)

15	10	14	5	0	0		15	15	11	0	0
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]		array[0]	array[1]	array[2]	array[3]	array[4]
		First Operation							Second Operation		

-10-5=-15 -> sin(-15*pi/180)

2	11	15	15	15	#
array[0]	array[1]	array[2]	array[3]	array[4]	array[5]

Operation with Pi (2+3.14)

*There are 4 overflows types:

1) $\sqrt{x}$ if x <0

2) result=x x>9999 or x<-999

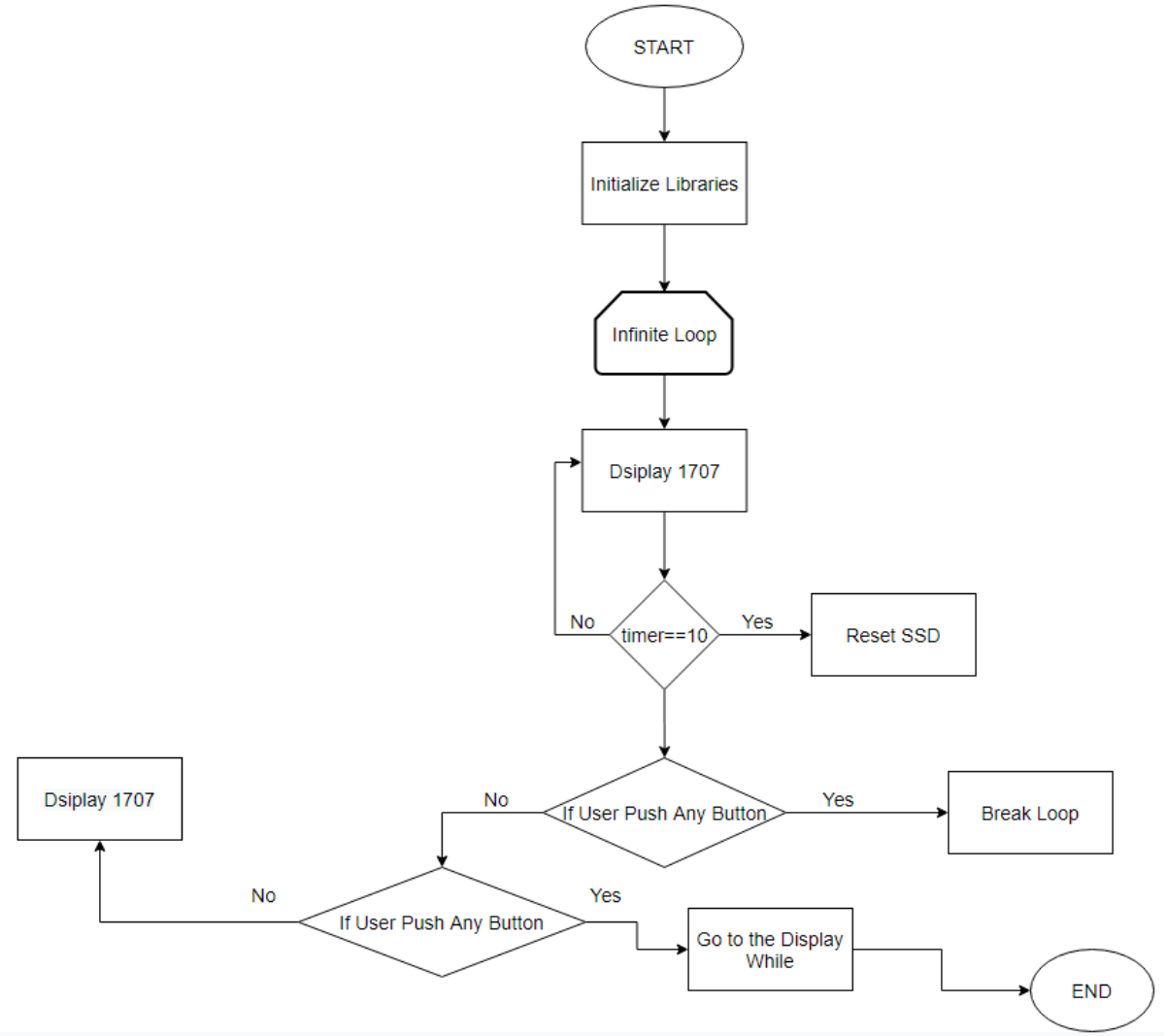
3)ln(-2)

4)x/0

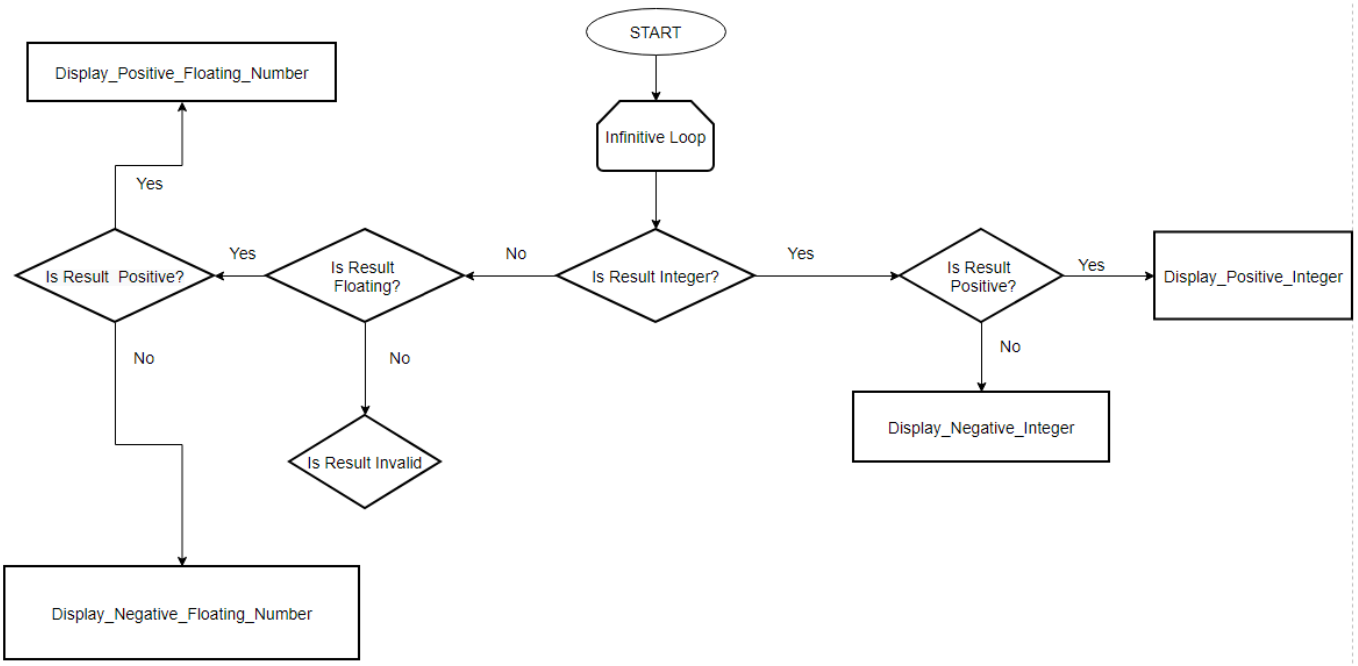
Also if user want to do sin(x/0*pi/180),this is also overflow.

3. Flowchart, BlockDiagram,Schematic,Codes:

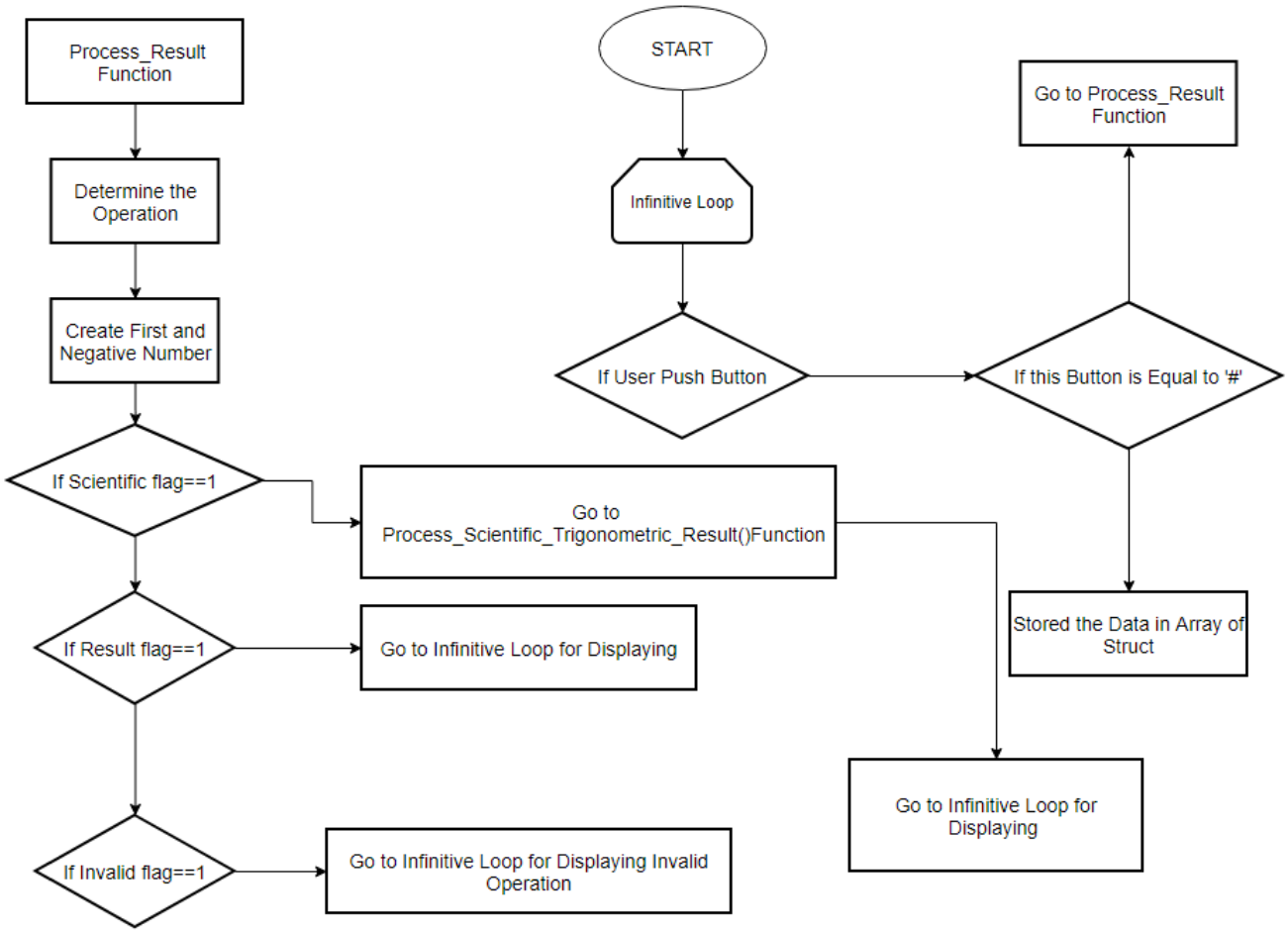
Below there are flowcharts for project2:



Flowchart-1.1 (First Part)

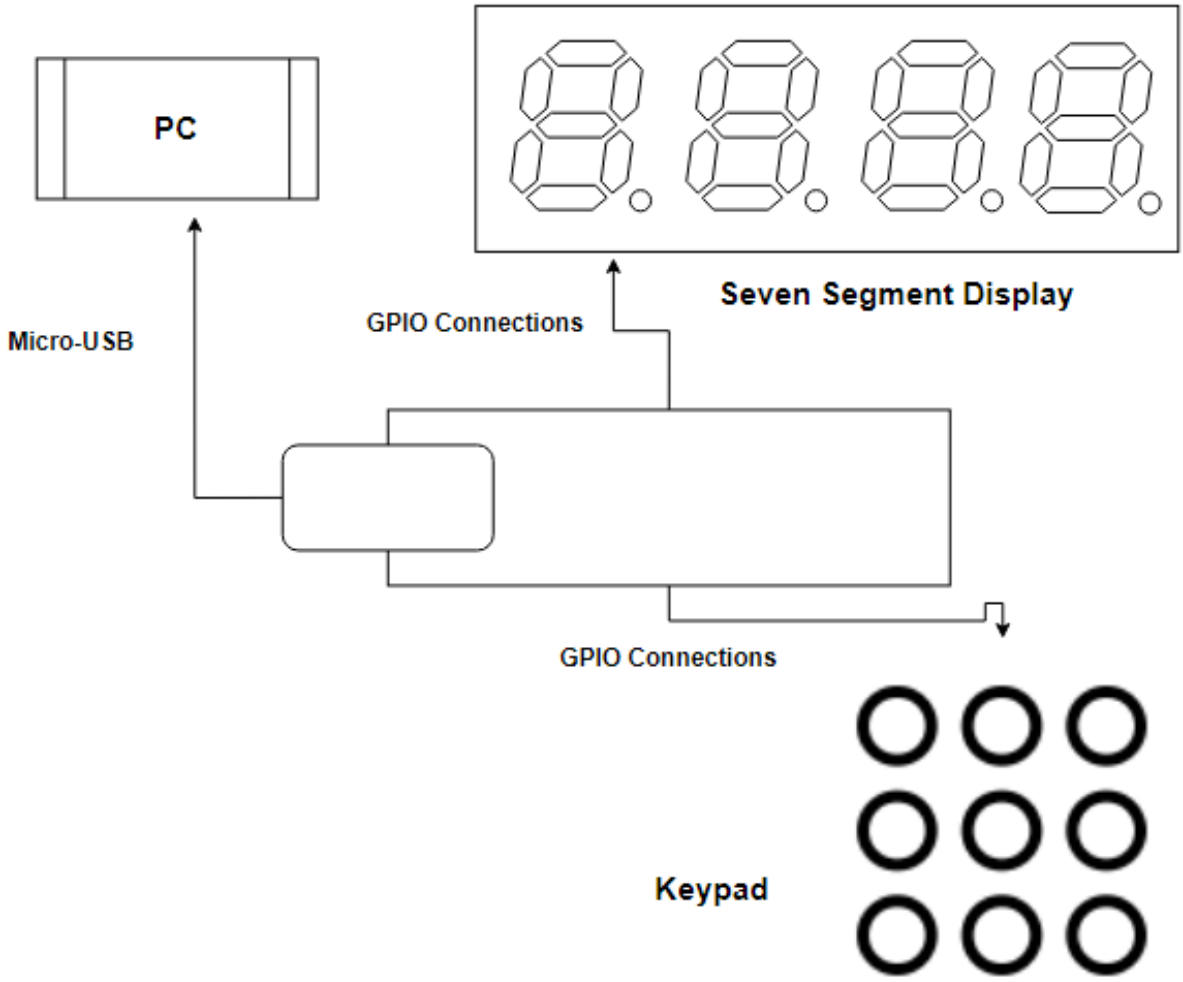


Flowchart-1.2(Second Part)



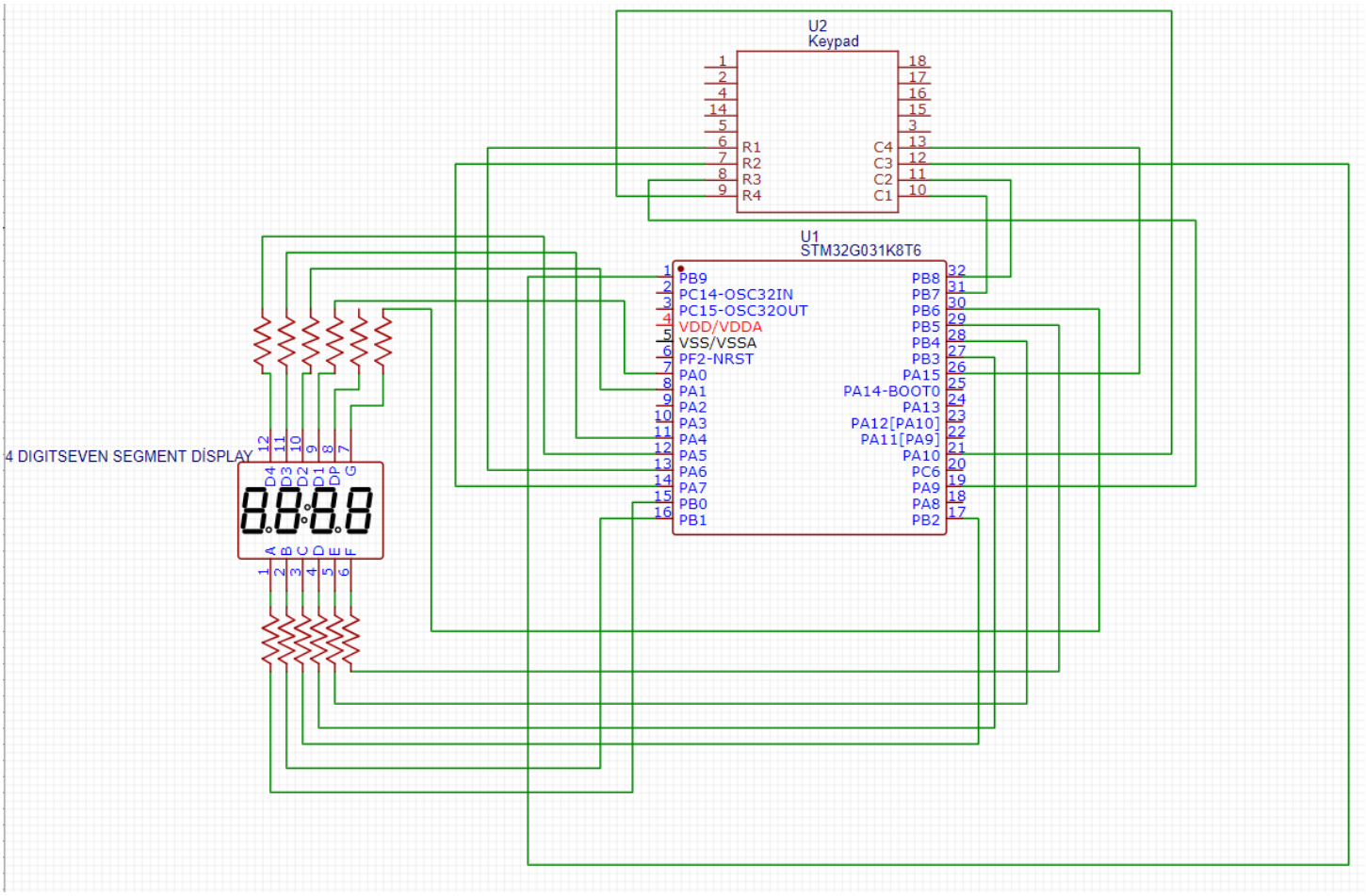
Flowchart-1.3(Third Part)

Below there is a blockdiagram for project2:



BlockDiagram-1

Below there is a Schematic for project2:



Schematic-1.1

Below there are Codes for project2:

```
/*
 * main.c
 *This project include 3 different libraries:
 *1)Keypad: have functions like that Keypad initializes,interrupt
 *2)SSD:have functions like thatinitializes,set_ssd,display_numbers(this numbers
can be nagative,positive, positive floating point format,negative floating point
format)
 *3)Process:All other operations in this library,for example calculation
operations,keypad_number_selection,or some initialize (timer,watchdog)
 *this calculator function have these operations types:
 *x+x
 *x-y
 *x+y
 *ln(x)
 *sin(x),cos(x),tan(x),cot(x),sin(x+y),cos(x+y),tan(x+y),cot(x+y) // variables
numbers can increase for example (x y z f)
 *User can this operation -> first operation ln(x) and then second operation
ln(ln(x)) or tan(ln(x)) // user can do any scientific operation after first
operation
 *User can this operation -> first operation ln(x) and then second operation
y+ln(x)  // user can do any normal operation after first operation
 *Also this code give a overflow signal like that(----),when -> result>9999
,result<-999 ,x/0,  (ln(x) -> x<0)
 *User can use negative number
 *User can see numbers like that 1.23, -1.23 ,11.23,-11.2,111.2,1111.
 *there are a lot of flags for some controls for example if user want to enter
negative number,or if result negative maybe
 *there are a lot of if-else statement because we have to check always a lot of
things for example is number negative? or integer? or overflow?
 * author: Omer Cebeci 171024007
 */

#include "stm32g0xx.h"
#include "SSD.h"
#include "Keypad.h"
#include "Process.h"



void TIM3_IRQHandler(){// this timer is used idle mode,if user dont push
anything into 10 seconds,ssd will close.
	if(timer<10){
		timer=timer+1;
	}
	TIM3->SR &=  ~(1U<<0);

}
```

```
int main(void) {
	result_old=0;
	System_Segment_Init(); // ssd initialize

	Keypad_Library_Init(); // keypad initialize

	Process_Init(); // timer initialize
	//Timer1_Init();
	//Delay(500);
	System_Decimal_Set();// ssd's decimal point is setted
	//Delay(500);
	//First_2_Last_2();
	//Delay(500);
	//TIM3->DIER=1;
	System_Keypad_Set();// R1 R2 R3 R4 are setted (in Keypad)
	while(idle_flag==0){ // for idle state if user dont push button along ten
second,ssd reset.
		if(timer<10){
			Delay(50);
			First_2_Last_2();// first four digit my school number display
			Delay(50);
		}
		if(timer==10){// check the time,if ten secons happpen reset ssd
			Delay(50);
			SSD_Reset();
			Delay(50);
		}
		if(idle_flag==1){// if user push one button ,break the while and
pass the calculation while...
			break;
		}
	}
timer=0;


while(1) {// calculation while

		if(timer==10){ // if user dont push '=' (in keypad '#'),along ten
seconds ssd display my school number...
			Delay(50);
			First_2_Last_2();// first four digit my school number display
			Delay(50);
			if(idle_flag!=1){// if user push one button ,break the while
and pass the calculation while...
					timer=11;
				}
		}
```

```
else {// if user push '=' (in keypad '#'),along ten seconds ,user can see
results...
                        double b ;
                        int  tantitive_result ;
                        result_old=result;
                        if(result_old>9999|| result_old<-999){ // this if
statement for overflow.../ if there are a overflow situation,every flag will be
zero ,only invalid flag set
                                Scientific_Trigonometric_flag=0;// we reset all
flag only invalid flag will set.
                                result_flag=0;//reset
                                invalid_flag=1;//set
                                result_old=0;//reset
                                result=0;//reset    // 2.24-  2=0.24
                        }
                        tantitive_result=(int)result_old;// tantitive result
for find the floating number...
                        if(Scientific_Trigonometric_flag==1){ //
Number_array_check condition is for second operation,if Number_array_check is
not equal zero we did same operation

                                        if(result_old-tantitive_result==0){ //
number is normal number there is no floating point

      if(result_old>0){  // if result is positive

                        Delay(50);

                        Number_Find(result_old);

                        Delay(50);
                                                                }
                                                                     else { //
if result is negative

                         result_old=result_old*(-1);

                         Negatif_Number_Find((uint32_t)result_old);

                         result_old=result_old*(-1);
                                                                }
                                        }
                                        else{// result-result_tantitive=> after
dat   1.25-1=0.25

                    b=b*1000;
```

```
            //Delay(50);

            Floating_Display(tantitive_result,(b)); // first argument is before
dat,second argument is after dat.

            //Delay(50);
                                                        }
                                                        else { // if result is
less than zero

            b=result_old-tantitive_result;

            b=b*1000*(-1);

            Floating_Display_Negative(tantitive_result,b);

            b=b*(-1);
                                                        }
                                        }

                          }
                         else if (result_flag==1){// if Number_array_check is
equal zero,we will do next operation so we dont display result.

                                                    if(result_old-
tantitive_result==0){ // check for floating or not ?

      if(result_old>0){  // if result is positive

            //Delay(10);

            Number_Find(result_old);

            //Delay(10);
                                                                    }
                                                                    else {
// if result is negative

             result_old=result_old*(-1);

             Negatif_Number_Find((uint32_t)result_old);

             result_old=result_old*(-1);
                                                                    }
                                                    }
                                                    else{

      if(result_old>0){// if result is bigger than zero

                          b=result_old-tantitive_result;

                          b=b*1000;

                          Floating_Display(tantitive_result,(b)); // first
argument is before dat,second argument is after dat.
                                                                }
```

```
                                                                    else { //
if result is less than zero
                            b=result_old-tantitive_result;
                            if(b<0)
                                  b=b*1000*(-1);
                            if(tantitive_result<0)
                                  tantitive_result=tantitive_result*(-1);

                            Floating_Display_Negative(tantitive_result,b);
                            //b=b*(-1);
                                                                        }
                                                  }


      }
                        else if (invalid_flag==1){// if Number_array_check is equal
zero,we will do next operation so we dont display result.
                                                                                //
Delay(50);

Display_zero();
                                                                                //
Delay(50);
                                    }
                                   else if(Is_there_new_Operation=1){// if first
operation finish and user want to do another operation,user will new number
values at ssd...
                                                                                //
Delay(50);


SSD_Number_Display();

                                                                                //
Delay(50);
                                              }

                        Watchdog_Update();

    }//for else
    }



    return 0;
}
```

```c
/*
 * Keypad.h
 *
 *  Created on: 18 Ara 2020
 *      Author: ÖMER
 */

#ifndef KEYPAD_H_
#define KEYPAD_H_

int idle_flag ;
int equal_check;
void Keypad_Library_Init(void);
void System_Keypad_Button_Init(void); // for keypad initialize
void System_Keypad_Output_Init(void);
void System_Keypad_Set(void);
void System_Keypad_Reset(void);


#endif /* KEYPAD_H_ */
```

```c
/*
 * Keypad.c
 *
 *  Created on: 18 Ara 2020
 *      Author: ÖMER
 */

#include "stm32g0xx.h"
#include "Keypad.h"
#include "SSD.h"
#include "Process.h"
idle_flag =0;// if idle flag is zero, user never push the button,
 equal_check=0; // if equla_check is zero user dont't push '='(in keypad like
that '#')
void Keypad_Library_Init(){
      System_Keypad_Output_Init();
      System_Keypad_Button_Init();
}
```

```c
void System_Keypad_Output_Init(){
       RCC->IOPENR |= (3U << 0); // GPIOA and GPIOB clock enable
       GPIOA->MODER &=  ~(3U<<2*6); //pa6 as output ,in the keypad R1
       GPIOA->MODER |=  (1U<<2*6);


       GPIOA->MODER &=  ~(3U <<2*7); //pa7  as output ,in the keypad R2
       GPIOA->MODER |=  (1 << 2*7);

       GPIOA->MODER &=  ~(3U <<2*9); //pa9  as output ,in the keypad R3
       GPIOA->MODER |=  (1 << 2*9);

       GPIOA->MODER &=  ~(3U <<2*10); //pa10  as output ,in the keypad R4
       GPIOA->MODER |=  (1 << 2*10);
}
void System_Keypad_Set(){
      GPIOA->ODR |= (1<<6);  //pa6 set,R1 set
      GPIOA->ODR |= (1<<7);  //pa7 set,R2 set
      GPIOA->ODR |= (1<<9);  //pa9 set,R3 set
      GPIOA->ODR |= (1<<10);  //pa10 set,R4 set

}
void System_Keypad_Reset(){
      GPIOA->ODR &=  ~(1U <<6); //pa6 reset,R1 reset
      GPIOA->ODR &=  ~(1U <<7); //pa7 reset,R2 reset
      GPIOA->ODR &=  ~(1U <<9); //pa9 reset,R3 reset
      GPIOA->ODR &=  ~(1U <<10);  //pa10 reset,R4 reset
}
void System_Keypad_Button_Init(){
       RCC->IOPENR |= (3U << 0); // GPIOA and GPIOB clock enable
      /*set pb7 as input */ // in keypad C1
            GPIOB->MODER &= ~(3U << 2*7);
            GPIOB->PUPDR |= (2U << 2*7);
      /*set pb8 as input */ // in keypad C2
            GPIOB->MODER &= ~(3U << 2*8);
            GPIOB->PUPDR |= (2U << 2*8);
      /*set pb9 as input */ // in keypad C3
            GPIOB->MODER &= ~(3U << 2*9);
            GPIOB->PUPDR |= (2U << 2*9);
      /*set pa15 as input */ // in keypad C4
            GPIOA->MODER &= ~(3U << 2*15);
            GPIOA->PUPDR |= (2U << 2*15);

      /* arrangement pb7 as interrupt*/
          EXTI->RTSR1 |=(1U<<7);
          EXTI->EXTICR[1] |= (1U<<8*3);
          EXTI->IMR1 |=(1<<7);
          NVIC_SetPriority(EXTI4_15_IRQn,0);
          NVIC_EnableIRQ(EXTI4_15_IRQn);

    /* arrangement pb8 as interrupt*/
          EXTI->RTSR1 |=(1U<<8);
          EXTI->EXTICR[2] |= (1U<<8*0);
          EXTI->IMR1 |=(1<<8);
          NVIC_SetPriority(EXTI4_15_IRQn,0);
            NVIC_EnableIRQ(EXTI4_15_IRQn);
```

```
      /* arrangement pb9 as interrupt*/
          EXTI->RTSR1 |=(1U<<9);
          EXTI->EXTICR[2] |= (1U<<8*1);
          EXTI->IMR1 |=(1<<9);
          NVIC_SetPriority(EXTI4_15_IRQn,0);
            NVIC_EnableIRQ(EXTI4_15_IRQn);


    /* arrangement pa15 as interrupt*/
          EXTI->RTSR1 |=(1U<<15);
          //EXTI->EXTICR[1] |= (1U<<8*0);
          EXTI->IMR1 |=(1<<15);
          NVIC_SetPriority(EXTI4_15_IRQn,0);
            NVIC_EnableIRQ(EXTI4_15_IRQn);

}
void EXTI4_15_IRQHandler (){

      Delay(1000); // for bouncing
      idle_flag=idle_flag+1;// if user push the button,this is for timer
interrupt
      Is_there_new_Operation=1;// it is send a message to ssd for display new
number
       // we have to reset this flag to clean ssd..
      // we have to reset these values for new operations
      //Scientific_Trigonometric_flag=0;
      //result_flag=0;


      System_Keypad_Reset();
      if((EXTI->RPR1>>7)&1){ // C1 Button is pressed (1,4,7,*)
            Process_C1_Check_Function();
            Scientific_Trigonometric_flag=0;
            result_flag=0;
            invalid_flag=0;
            EXTI->RPR1 |=(1<<7);
      }
      else if((EXTI->RPR1>>8)&1){  // c2 button is pressed (2,5,8,0)
            Process_C2_Check_Function();
            Scientific_Trigonometric_flag=0;
            result_flag=0;
            invalid_flag=0;
            EXTI->RPR1 |=(1<<8);
      }
      else if((EXTI->RPR1>>9)&1){// c3 button is pressed (3,6,9,#)
            Process_C3_Check_Function();
            Delay(1000);
            EXTI->RPR1 |=(1<<9);
      }

      else if((EXTI->RPR1>>15)&1){ // c4 button is pressed (a,b,c,d)
            Process_C4_Check_Function();
            Scientific_Trigonometric_flag=0;
            result_flag=0;
            invalid_flag=0;
            EXTI->RPR1 |=(1<<15);
            }
```

```
	System_Keypad_Set();
	Watchdog_Update();
}
```

```
/*
 * Process.h
 *
 *  Created on: 18 Ara 2020
 *      Author: ÖMER
 */

#ifndef PROCESS_H_
#define PROCESS_H_

int timer;
int operation_type ; // 1->addition,2->substraction,3->mul. ,4->division
int operation_place;// this values is used for calculation first and second
number//// before operation_place,there is first number, after
operation_place,there is second number
int flag_timer;
int key_flag; // if key_flag is equal 0, calculator expect first number,if
key_flag is equal 1 ,calculator expect second number....
double result ; // result of the calculation
//double number_number[5];
double result_old; // for use the previous result
int result_flag ;
int Scientific_Trigonometric_flag;
int Is_there_new_Operation ;
typedef struct Number {
	double array_number[10];// this is our stored array ,all inputs come this
array and then  numbers are created and operation_type is determined
}Number;
Number number;
int Number_array_check;//The number of presses is kept in this value
int invalid_flag;

void Timer1_Init();
void First_2_Last_2();
void Display_School_Number();
void System_Timer_Init();
void Process_Init();
void Watchdog_Init();
void Watchdog_Update();

void Process_Scientific_Trigonometric_Result();
void Process_C1_Check_Function();
void Process_C2_Check_Function() ;
void Process_C3_Check_Function();
void Process_C4_Check_Function();
void Process_Result();


void SSD_Number_Display();
#endif /* PROCESS_H_ */
```

```c
/*
 * Process.c
 *
 *  Created on: 18 Ara 2020
 *      Author: ÖMER
 */

#include "stm32g0xx.h"
#include "SSD.h"
#include "Keypad.h"
#include "Process.h"
#include "math.h"

timer=0;
int operation_place =0 ;operation_type; // 1->addition,2->substraction,3->mul.
,4->division
int flag_timer =0;
int key_flag=0; // if key_flag is equal 0, calculator expect first number,if
key_flag is equal 1 ,calculator expect second number....
// result of the calculation,I dont give first value for result....
 result_flag=0 ;
Scientific_Trigonometric_flag=0;
double number_number[5];
invalid_flag=0; // if operation is invalid,invalid_flag would be 1, for example
(3/0) or sqrt(-2)
Is_there_new_Operation=1;
int Number_array_check =0;

void Process_Init(){
	//SystemCoreClockUpdate();
	System_Timer_Init();
	//Timer1_Init();
	//Watchdog_Init();
	//SysTick_Config(SystemCoreClock/1000);
}
void Timer1_Init(){
	RCC->APBENR2 |=(1u<<11);
	TIM1->CR1=0;
	TIM1->CR1 |= (1<<7);
	TIM1->CNT =0;
	TIM1->PSC =999; // every  one second
	TIM1->ARR=(16000);
	TIM1->DIER |=(1<<0);
	TIM1->CR1 =0;
	NVIC_SetPriority(TIM1_BRK_UP_TRG_COM_IRQn,0);
	NVIC_EnableIRQ(TIM1_BRK_UP_TRG_COM_IRQn);
}
```

```
void System_Timer_Init(){
            RCC->APBENR1 |=(1U<<1);
            TIM3->CR1=0;
            TIM3->CR1 |=(1<<7);
            TIM3->CNT =0;
            TIM3->PSC=999;  // old value 0
            TIM3->ARR=(16000); // less than 1 second  // old value 8000
            TIM3->DIER |=(1<<0);
            TIM3->CR1 |= (1<<0);
            NVIC_SetPriority(TIM3_IRQn,2);
            NVIC_EnableIRQ(TIM3_IRQn);

}
void Watchdog_Update(){
      IWDG->KR=0xAAAA;
}
void Watchdog_Init(){
      IWDG->KR=0x5555;
      IWDG->PR=1;
      //IWDG->RLR=0xFFF;
      //IWDG->WINR=0x0;
      IWDG->KR=0xCCCC;
}


void SysTick_Handler(){
      if(result_flag==1){
                           Number_Find(result);
                }
                else{
                           Delay(50);
                           SSD_Number_Display();
                           Delay(50);

                }

}
void First_2_Last_2(){

            Delay(500);
            System_D1();
            Display_Number(1);

            Delay(500);
            System_D2();
            Display_Number(7);

            Delay(500);
            System_D3();
            Display_Number(0);

            Delay(500);
            System_D4();
            Display_Number(7);



}
```

```
void Display_School_Number(){
	for(int i=0;i<500;i++){
			Delay(500);
			System_D1();
			Display_Number(4);

			Delay(500);
			System_D2();
			Display_Number(0);

			Delay(500);
			System_D3();
			Display_Number(0);

			Delay(500);
			System_D4();
			Display_Number(7);

		}
}
void Process_C1_Check_Function(){
	GPIOA->ODR |=(1<<6);  //R1 Check
	if((GPIOB->IDR>>7)&1){ // if it is true ,user push the one;
				number.array_number[Number_array_check]=1;
				Number_array_check=Number_array_check+1;

	}
	else{
			GPIOA->ODR &=  ~(1U <<6);// user dont push '1' (R1)
			GPIOA->ODR |=(1<<7);  //R2 Check
			if((GPIOB->IDR>>7)&1){ // if it is true ,user push the '4';
				number.array_number[Number_array_check]=4;
				Number_array_check=Number_array_check+1;


			}

			else {
				GPIOA->ODR &=  ~(1U <<7);// user dont push '4'
	(R2)
				GPIOA->ODR |=(1<<9);  //R3 Check  PA4
				if((GPIOB->IDR>>7)&1){ // if it is true ,user push the
'7';
				number.array_number[Number_array_check]=7;
				Number_array_check=Number_array_check+1;

				}
				else{
				GPIOA->ODR &=  ~(1U <<9);// user dont push '7'
	(PA4)
				GPIOA->ODR |=(1<<10);  //R4 Check
				if((GPIOB->IDR>>7)&1){ // if it is true ,user
push the '*'; // this is 'E' for scientific mode

	number.array_number[Number_array_check]=15;
					Number_array_check=Number_array_check+1;
				}
```

```
                }

        }
}
void Process_C2_Check_Function(){
        GPIOA->ODR |=(1<<6);  //R1 Check
        if((GPIOB->IDR>>8)&1){ // if it is true ,user push the '2';
                number.array_number[Number_array_check]=2;
                Number_array_check=Number_array_check+1;

                }

        else{
                        GPIOA->ODR &=  ~(1U <<6);// user dont push '2' (R1)
                        GPIOA->ODR |=(1<<7);  //R2 Check
                        if((GPIOB->IDR>>8)&1){ // if it is true ,user push the '5';
                                number.array_number[Number_array_check]=5;
                                Number_array_check=Number_array_check+1;

                        }

                        else {
                                        GPIOA->ODR &=  ~(1U <<7);// user dont push '5'
        (R2)
                                        GPIOA->ODR |=(1<<9);  //R3 Check  PA4
                                        if((GPIOB->IDR>>8)&1){ // if it is true ,user
push the '8';

        number.array_number[Number_array_check]=8;
                                                Number_array_check=Number_array_check+1;

                                        }

                                        else{
                                                GPIOA->ODR &=  ~(1U <<9);// user dont
push '8'     (PA4)
                                                GPIOA->ODR |=(1<<10);  //R4 Check
                                                if((GPIOB->IDR>>8)&1){ // if it is true
,user push the '0';

        number.array_number[Number_array_check]=0;

        Number_array_check=Number_array_check+1;

                                                }

                                        }

                        }

                }
}
```

```
void Process_C3_Check_Function(){
	GPIOA->ODR |=(1<<6);  //R1 Check
	if((GPIOB->IDR>>9)&1){ // if it is true ,user push the '3';
		number.array_number[Number_array_check]=3;
		Number_array_check=Number_array_check+1;


	}
	else{
			GPIOA->ODR &=  ~(1U <<6);// user dont push '3' (R1)
			GPIOA->ODR |=(1<<7);  //R2 Check
			if((GPIOB->IDR>>9)&1){ // if it is true ,user push the '6';
				number.array_number[Number_array_check]=6;
				Number_array_check=Number_array_check+1;

			}

			else {
					GPIOA->ODR &=  ~(1U <<7);// user dont
push '6'     (R2)
					GPIOA->ODR |=(1<<9);  //R3 Check  PA4
					if((GPIOB->IDR>>9)&1){ // if it is true
,user push the '9';

	number.array_number[Number_array_check]=9;

	Number_array_check=Number_array_check+1;

					}
					else{
							GPIOA->ODR &=  ~(1U <<9);//
user dont push '9'  (PA4)
							GPIOA->ODR |=(1<<10);  //R4
Check
							if((GPIOB->IDR>>9)&1){ // if
it is true ,user push the '#'; // for '='
								Process_Result();  //
later we will add mode variable for scientific ...

								timer=11;// if user
push '=' (in kaypad '#')
							}

					}

			}

	}

}
```

```
void Process_C4_Check_Function(){
	GPIOA->ODR |=(1<<6);  //R1 Check
	if((GPIOA->IDR>>15)&1){ // if it is true ,user push the 'A'; // addition
			number.array_number[Number_array_check]=11;
			Number_array_check=Number_array_check+1;
	}
	else{
			GPIOA->ODR &=  ~(1U <<6);// user dont push 'A' (R1)
			GPIOA->ODR |=(1<<7);  //R2 Check
			if((GPIOA->IDR>>15)&1){ // if it is true ,user push the 'B';
//substraction
				number.array_number[Number_array_check]=12;
				Number_array_check=Number_array_check+1;
			}
			else {
				GPIOA->ODR &=  ~(1U <<7);// user dont push 'B'
	(R2)
				GPIOA->ODR |=(1<<9);  //R3 Check  PA4
				if((GPIOA->IDR>>15)&1){ // if it is true ,user push
the 'C'; // multiplicatiıon
					number.array_number[Number_array_check]=13;
					Number_array_check=Number_array_check+1;
				}
				else{
					GPIOA->ODR &=  ~(1U <<9);// user dont push 'C'
	(PA4)
					GPIOA->ODR |=(1<<10);  //R4 Check
					if((GPIOA->IDR>>15)&1){ // if it is true ,user
push the 'D'; // division

	number.array_number[Number_array_check]=14;
						Number_array_check=Number_array_check+1;
					}

				}

			}

		}


}
```

```
void Process_Result(){
	Is_there_new_Operation=0;
	result=0;
	int i =0;
	//first number is found // 1 2 opetaitontype 2 3 #   2+2+2 2+2=4 +2 =6
	for(int i=0;i<10;i++){

	if((number.array_number[i]==11)||(number.array_number[i]==12)||(number.arr
ay_number[i]==13)||(number.array_number[i]==14)||(number.array_number[i]==15)){
			if(i==0 && number.array_number[0]==15
&&((number.array_number[1]!=11)&(number.array_number[1]!=12)&(number.array_numbe
r[1]!=13)&(number.array_number[1]!=14)&(number.array_number[1]!=15))){

	operation_place=operation_place+1;// ignore This if for 1+2=3 and + 5 = 8
					}
			else
if(number.array_number[0]==15&&((number.array_number[1]==11)||(number.array_numb
er[1]==12)||(number.array_number[1]==13)||(number.array_number[1]==14))){
				// this else if for 1+2 =3 and then second operation
ln(3)
				operation_place=operation_place+2; // because 15 11
(2)  15 and 11 is not a number value
				operation_type=15; // E '*'
				break ;
			}
					else{
					operation_type=number.array_number[i];
					break;
					}

		}
		else
			operation_place=operation_place+1;

	}

	if(operation_type==15){// For scientific calculations. if operations type
is equla 15,user push the button 'E'
		Scientific_Trigonometric_flag=1;

	}
	if(operation_place==0){ // if this condition is true,we have to use
previous result
			number_number[0]=result_old;// new first number is old
result.
	}
	else{// if this condition is not true,we have to produce new number...
		if(number.array_number[0]==15){// if this condition is true
,negative number will come
			for(i =1;i<operation_place;i++){

	number_number[0]=number_number[0]+pow(10,i-
1)*number.array_number[operation_place-i]*(-1);
			}
		}
		else
```

```
            for(i =0;i<operation_place;i++){

      number_number[0]=number_number[0]+pow(10,i)*number.array_number[operation_
place-i-1];
            }


      }

      //Second number is found
      int j =0;
      for(int i =(operation_place+1);i<10;i++){ // we find ,how many number
after operation place,these numbers comprise our second number
            if(number.array_number[i]==0)
                  break;
            else
                  j=j+1;
      }
      int m=0;
      for(int k=(j+operation_place) ; k>operation_place; --k){

      number_number[1]=number_number[1]+pow(10,(m))*number.array_number[k];
                  m=m+1;
            }
      switch (operation_type){// in this switch we select an operation type...
                  case 11:// for addition
                        for(int i=0;i<5;i++){
                              result=result+number_number[i];
                        }
                  break;

                  case 12: // substraction
                        result =number_number[0]-number_number[1];

                  break ;

                  case 13://(*)
                        result =number_number[0]*number_number[1];
                  break ;

                  case 14: // (/) division
                        if(number_number[1]==0)
                              invalid_flag=1;
                        result=number_number[0]/number_number[1];
                  break ;
      }
      if(Scientific_Trigonometric_flag==1){ // for scientific operations
            Process_Scientific_Trigonometric_Result();// if user push relevent
buttons,pc go to the this function for other operations(log,ln,sin,cos,tan,cot)
      }
      else if(invalid_flag==0)// if operation is invalid this flag set for
display '----'
            result_flag=1; // for normal operations
// for next calculation
Number_array_check=0;
```

```
for(int i =0;i<10;i++)
	number.array_number[i]=0;

number_number[0]=0;
number_number[1]=0;
operation_place=0;
operation_type=0;

//////////
}
void Process_Scientific_Trigonometric_Result(){
	int i =0;
	int j=0;
	number_number[0]=0,number_number[1]=0 ; // clear the numbers in stored
array

	for( i =0;i<10;i++){
		if(number.array_number[i]==15&&number.array_number[i+1]==11){// EA
for log
			operation_type= 26 ; // 15+11
			break ;
		}
		else if (number.array_number[i]==15 &&
number.array_number[i+1]==12){ // EB for ln
			operation_type= 27 ; // 15+12
			break ;
		}

		else if (number.array_number[i]==15 &&
number.array_number[i+1]==13){// EC for sqrt
			operation_type= 28 ; // 15+13
			break ;
		}
		else if (number.array_number[i]==15 &&
number.array_number[i+1]==14){// ED for x^2
					operation_type= 29 ; // 15+14
					break ;
				}
		else if (number.array_number[i]==15&&number.array_number[i+1]==15){
// trigonometric mode
			if(number.array_number[i+2]==11){ // sinus
				operation_type= 41 ; // 15+15+11
				break ;
			}
			else if(number.array_number[i+2]==12){// cosinus
				operation_type= 42 ; // 15+15+12
				break ;
			}
			else if(number.array_number[i+2]==13){ //tan
				operation_type= 43 ; // 15+15+13
				break ;
			}
			else if (number.array_number[i+2]==14){ // cot
				operation_type= 44 ; // 15+15+14
				break ;
			}
		}
	}
```

```
	if(number.array_number[0]==15 &&
((number.array_number[1]!=11)&(number.array_number[1]!=12)&(number.array_number[
1]!=13)&(number.array_number[1]!=14))){// if this condition is true ,negative
number will come
				for(i =1;i<operation_place;i++){

	number_number[0]=number_number[0]+pow(10,i-
1)*number.array_number[operation_place-i]*(-1);
				}
			}
	else if (number.array_number[0]==15 &&
((number.array_number[1]==11)||(number.array_number[1]==12)||(number.array_numbe
r[1]==13)||(number.array_number[1]==14))){
		// this statement for 1+2=3 and then ln(3)
		number_number[0]=result_old;
	}
			else// this statement for 1+2 and then A BC D that means 3+x
,3-x,3/5,3*3

			for(i =0;i<operation_place;i++){

	number_number[0]=number_number[0]+pow(10,i)*number.array_number[operation_
place-i-1];
			}
	switch (operation_type){
				case 26:
					result=(double)log10(number_number[0]);
					break;
				case 27:
					result=(double)log(number_number[0]);
					break ;
				case 28:
					if(number_number[0]<0){
						invalid_flag=1;
						Scientific_Trigonometric_flag=0;
						break;
					}
					else{
						result=(double)sqrt(number_number[0]);
						break ;
					}
				case 29:
					result=number_number[0]*number_number[0];
					break ;
				case 41:
					result=(double)sin((number_number[0])*3.14/180);
					break ;
				case 42:
					result=(double)cos((number_number[0])*3.14/180);
					break ;
				case 43:
					result=(double)tan((number_number[0])*3.14/180);
					break ;
				case 44:

	result=(double)(1/(tan((number_number[0])*3.14/180)));
					break ;
		}
```

```
}

void SSD_Number_Display(){
	if(Number_array_check==1){// if number_array_check is equal zero first
number come,so we should write digit one
		Delay(100);
		System_D4();
		Display_Number(number.array_number[0]);// if number_array_check is
equal zero,only first of array have number.

	}
	else if (Number_array_check==2){
		Delay(100);
		System_D3();
		Display_Number(number.array_number[0]);

		Delay(100);
		System_D4();
		Display_Number(number.array_number[1]);

	}
	else if(Number_array_check==3){
		Delay(100);
		System_D2();
		Display_Number(number.array_number[0]);

		Delay(100);
		System_D3();
		Display_Number(number.array_number[1]);

		Delay(100);
		System_D4();
		Display_Number(number.array_number[2]);
	}
	else if(Number_array_check==4){
		Delay(100);
		System_D1();
		Display_Number(number.array_number[0]);

		Delay(100);
		System_D2();
		Display_Number(number.array_number[1]);

		Delay(100);
		System_D3();
		Display_Number(number.array_number[2]);

		Delay(100);
		System_D4();
		Display_Number(number.array_number[3]);

	}
}
```

```c
/*
 * SSD.h
 *
 *  Created on: 18 Ara 2020
 *      Author: ÖMER
 */

#ifndef SSD_H_
#define SSD_H_

void System_Segment_Init();


void System_Number1();
void System_Number2();
void System_Number3();
void System_Number4();
void System_Number5();
void System_Number6();
void System_Number7();
void System_Number8();
void System_Number9();
void System_Number0();
void System_NumberA();
void System_NumberB();
void System_NumberAC();
void System_NumberD();


void System_Decimal();
void System_D1();
void System_D2();
void System_D3();
void System_D4();
void SSD_Reset();


void Negatif_Number_Find(uint32_t);
void Number_Find(uint32_t);
void Display_Number(int);
void Display_zero();
void Floating_Display(uint16_t,uint16_t);
void Floating_Display_Negative(uint16_t,double);
void System_Decimal_Reset();
void System_Decimal_Set();
void Delay(volatile uint32_t );



#endif /* SSD_H_ */
```

```c
/*
 * SSD.c
 *
 *  Created on: 18 Ara 2020
 *      Author: ÖMER
 */
#include "stm32g0xx.h"
#include "SSD.h"
#include "Keypad.h"

void Display_zero(){// if invalid operation comprise,ssd display '----', this
function print ---- for overflow and invalid operation

            Delay(500);
            System_D1();
            Display_Number(12);

            Delay(500);
            System_D2();
            Display_Number(12);

            Delay(500);
            System_D3();
            Display_Number(12);

            Delay(500);
            System_D4();
            Display_Number(12);
}
void Display_Number(int a){// a is our digit,this if-else statement ,for display
digit

      if(a==0)
            System_Number0();
      else if (a==1)
            System_Number1();
      else if (a==2)
                  System_Number2();
      else if (a==3)
                  System_Number3();
      else if (a==4)
                  System_Number4();

      else if (a==5)
                  System_Number5();
      else if (a==6)
                  System_Number6();
      else if (a==7)
                  System_Number7();
      else if (a==8)
                  System_Number8();
      else if (a==9)
                        System_Number9();
      else if (a==11) // addition
                  System_NumberA();
```

```
	else if(a==12||a==15)
			System_NumberB();
	else if(a==13)
			System_NumberC();
	else if(a==14)
			System_NumberD();


}
void Negatif_Number_Find(uint32_t number){
	uint32_t binler,yuzler,onlar;
	uint32_t number_t;
	for(int i=0;i<2;i++){
			number_t=number;
			//number_t=number_t%0;
			//binler=(number_t/1000); // we cant display -1000 we can
display only -999

			Delay(50);
			System_D1();
			Display_Number(12); // for '-'

			number_t=number%1000; // find hunders digits
			yuzler=(number_t /100);

			Delay(50);
			System_D2();
			Display_Number(yuzler);

			number_t=number_t%100;  // find tens digits
			onlar=number_t/10;

			Delay(50);
			System_D3();
			Display_Number(onlar);

			number_t=number_t%10;  // find birler digits

			Delay(50);
			System_D4();
			Display_Number(number_t);
		}


}

void Number_Find(uint32_t number){// this function find the number's digit
	uint32_t binler,yuzler,onlar;
	uint32_t number_t;
	for(int i=0;i<2;i++){
		number_t=number;
		//number_t=number_t%0;
		binler=(number_t/1000);

		Delay(30);
		System_D1();
```

```
            Display_Number(binler);

            number_t=number%1000; // find hunders digits
            yuzler=(number_t /100);

            Delay(30);
            System_D2();
            Display_Number(yuzler);

            number_t=number_t%100;  // find tens digits
            onlar=number_t/10;

            Delay(30);
            System_D3();
            Display_Number(onlar);

            number_t=number_t%10;  // find birler digits

            Delay(30);
            System_D4();
            Display_Number(number_t);

      }
}
void Floating_Display_Negative(uint16_t a, double b ){ // a is number that
before dat ,b is number that after dat.
      int number_t ;
            double binler,yuzler,onlar;
            if(a<10){ // number will be like that 1.002  d1's decimal will be
open
                        //for(int i=0;i<2;i++){
                              Delay(50);
                              System_Decimal_Set();

                              System_D1();
                              Display_Number(12); // for '-'
                              // add decimal..
                               // decimal is written in D1 digit
                              //////////

                              Delay(50);
                              System_D2();
                              Display_Number(a); // the number that before the
dat(.)
                              System_Decimal_Reset();
                              number_t=b; // find hunders digits
                              yuzler=(number_t /100);

                              Delay(50);
                              System_Decimal_Set();

                              System_D3();
                              Display_Number(yuzler);
```

```
                                number_t=number_t%100;  // find tens digits
                                onlar=number_t/10;

                                Delay(50);
                                System_Decimal_Set();

                                System_D4(); //Note : we dont display ones
digits...
                                Display_Number(onlar);


                        //}
                }

                else if (a<99){ // number will be like that 10.222  d2's decimal
will be open
                        for(int i=0;i<2;i++){
                                                System_Decimal_Set();
                                                Delay(50);
                                                System_D1();
                                                Display_Number(12); // for '-'

                                                // add decimal..
                                                // decimal is written in D1 digit

                                                number_t=a;
                                                onlar=number_t/10;

                                                System_Decimal_Set();
                                                Delay(50);
                                                System_D2();
                                                Display_Number(onlar);


                                                number_t=number_t%10;  // find ones
digit


                                                Delay(50);
                                                System_D3();
                                                Display_Number(number_t);
                                                System_Decimal_Reset();
                                                // add decimal ....
                                                 // decimal is written in D2 digit
                                                ////////


                                                number_t=b;
                                                binler=0,yuzler=0,onlar=0;

                                                yuzler=number_t/100;

                                                System_Decimal_Set();
                                                Delay(50);
                                                System_D4();
                                                Display_Number(yuzler);
```

```
                                }

            }
            else if(a<999){  // number will be like that 102.222 d3's decimal
will be open
                  for(int i =0;i<2;i++){
                        System_Decimal_Set();
                        Delay(50);
                        System_D1();
                        Display_Number(12); // for '-'


                        number_t=a;
                         // find hunders digits
                        yuzler=(number_t /100);

                        System_Decimal_Set();
                        Delay(50);
                        System_D2();
                        Display_Number(yuzler);
                        number_t=number_t%100;
                        onlar=number_t/10; // find tens digit

                        System_Decimal_Set();
                        Delay(50);
                        System_D3();
                        Display_Number(onlar);
                        number_t=number_t%10;  // find ones  digit

                        Delay(50);
                        System_D4();
                        Display_Number(number_t);
                        System_Decimal_Reset();
                        // there is no place for b(after dot)
                  }

            }
            else{
                        Number_Find(a);// if b is bigger than 999,ssd'S four
digits display for b's digits
                  }
      binler=0,yuzler=0,onlar=0,number_t=0;
      }
```

```
void Floating_Display(uint16_t a, uint16_t b){// first argument is before
dat,second argument is after dat.
      int number_t ;
      double binler,yuzler,onlar;
      if(a<10){ // number will be like that 1.002  d1's decimal will be open
2.111   22.22  223.
                  for(int i=0;i<1;i++){

                        Delay(100);
                        System_D1();
                        Display_Number(a);
                        System_Decimal_Reset();
                        // add function for decimal..
                         // decimal is written in D1 digit
                        //////////

                        number_t=b; // find hunders digits
                        yuzler=(number_t /100);

                        System_Decimal_Set();
                        Delay(100);
                        System_D2();
                        Display_Number(yuzler);



                        number_t=number_t%100;  // find tens digits
                        onlar=number_t/10;

                        System_Decimal_Set();
                        Delay(100);
                        System_D3();
                        Display_Number(onlar);
                        // add function for decimal..


                        number_t=number_t%10;  // find birler digits

                        System_Decimal_Set();
                        Delay(100);
                        System_D4();
                        Display_Number(number_t);
                        // add function for decimal..
                        //Delay(100);
            }
      }

      else if (a<99){ // number will be like that 10.222  d2's decimal will be
open
            for(int i=0;i<2;i++){
                                    number_t=a;
                                    onlar=number_t/10;

                                    System_Decimal_Set();
                                    Delay(50);
                                    System_D1();
                                    Display_Number(onlar);
```

```
                                        number_t=number_t%10;  // find birler
digits


                                        Delay(50);
                                        System_D2();
                                        Display_Number(number_t);
                                        System_Decimal_Reset();
                                        // add decimal ....
                                         // decimal is written in D2 digit
                                        ////////

                                        number_t=b;
                                        binler=0,yuzler=0,onlar=0;
                                        /*
                                        //number_t=b%1000; // find hunders digits
                                        yuzler=(number_t /100);

                                        Delay(50);
                                        System_D3();
                                        Display_Number(yuzler);
                                        */

                                          // find hundreds digits
                                        yuzler=number_t/100;
                                        System_Decimal_Set();
                                        Delay(50);
                                        System_D3();
                                        Display_Number(yuzler);

                                        number_t=number_t%100;  // find tens
digits
                                        onlar=number_t/10;

                                        System_Decimal_Set();
                                        Delay(50);
                                        System_D4();
                                        Display_Number(onlar);

                           }

      }
      else if(a<999){  // number will be like that 102.222 d3's decimal will be
open
             for(int i =0;i<2;i++){
                    number_t=a;
                     // find hunders digits
                    yuzler=(number_t /100);

                    System_Decimal_Set();
                    Delay(50);
                    System_D1();
                    Display_Number(yuzler);

                    number_t=number_t%100;
                    onlar=number_t/10; // find tens digit

                    System_Decimal_Set();
                    Delay(50);
                    System_D2();
                    Display_Number(onlar);
```

```
                    number_t=number_t%10;  // find ones  digit

                    Delay(50);

                    System_D3();
                    Display_Number(number_t);
                    System_Decimal_Reset();
                    // add decimal ...
                     // decimal is written in D3 digit

                    number_t=b/100;
                    System_Decimal_Set();
                    Delay(50);
                    System_D4();
                    Display_Number(number_t);

             }

      }
      else{
            Number_Find(a);// if b is bigger than 999,ssd'S four digits display
for b's digits
      }
binler=0,yuzler=0,onlar=0,number_t=0;
}



void System_Segment_Init(){
       RCC->IOPENR |= (3U << 0); // GPIOA and GPIOB clock enable

       GPIOA->MODER &=  ~(3U<<2*0); //pa0
       GPIOA->MODER |=  (1U<<0);


       GPIOA->MODER &=  ~(3U <<2*1); //pa1  as output
       GPIOA->MODER |=  (1 << 2);

       GPIOA->MODER &=  ~(3U <<2*4); //pa4  as output
       GPIOA->MODER |=  (1 << 8);

       GPIOA->MODER &=  ~(3U <<2*5); //pa5  as output
       GPIOA->MODER |=  (1 << 10);

       GPIOA->MODER &=  ~(3U <<2*6); //pa6  as output
       GPIOA->MODER |=  (1 << 2*6);

       GPIOA->MODER &=  ~(3U <<2*11); //pa11  as output
       GPIOA->MODER |=  (1 << 2*11);

       GPIOB->MODER &=  ~(3U << 0); //pB0  output
       GPIOB->MODER |=  (1 << 0);

       GPIOB->MODER &=  ~(3U << 2*1); //pB1  output
       GPIOB->MODER |=  (1 << 2);

       GPIOB->MODER &=  ~(3U << 2*2); //pB2  output
       GPIOB->MODER |=  (1 << 4);

       GPIOB->MODER &=  ~(3U <<2* 3); //pB3  output
       GPIOB->MODER |=  (1 << 6);
```

```
        GPIOB->MODER &=  ~(3U <<2* 4); //pB4  output
        GPIOB->MODER |=  (1 << 8);

        GPIOB->MODER &=  ~(3U <<2* 5); //pB5  output
        GPIOB->MODER |=  (1 << 10);

        GPIOB->MODER &=  ~(3U <<2* 6); //pB6  output
        GPIOB->MODER |=  (1 << 12);
}


void System_Number1(){
       GPIOB->ODR =0x79  ;
}
void System_Number2(){

       GPIOB->ODR =0xA4  ;
}
void System_Number3(){

       GPIOB->ODR =0xB0  ;
}
void System_Number4(){

       GPIOB->ODR =0x99  ;
}
void System_Number5(){

       GPIOB->ODR =0x92  ;
}
void System_Number6(){

       GPIOB->ODR =0x2  ;
}
void System_Number7(){

       GPIOB->ODR =0xF8  ;
}
void System_Number8(){
       GPIOB->ODR =0  ;
}
void System_Number9(){
       GPIOB->ODR =0x90  ;
}
void System_Number0(){
       GPIOB->ODR =0x40  ;
}
void System_NumberA(){ // for addition
       GPIOB->ODR =0x8  ;
}
void System_NumberB(){ // For substraction
       GPIOB->ODR =0x3F  ;
}
void System_NumberC(){ // For substraction
       GPIOB->ODR =0x4E  ;
}
void System_NumberD(){ // For Division
       GPIOB->ODR =0x4F  ;
}
```

```c
void System_D1(){
      GPIOA->ODR |=(1<<0); //pa0 (1)
      GPIOA->ODR &= ~(1U <<1);
      GPIOA->ODR &= ~(1U <<4);
      GPIOA->ODR &= ~(1U <<5);
}
void System_D2(){
      GPIOA->ODR |=(1<<1); //PA1  (2)
      GPIOA->ODR &= ~(1U <<0);
      GPIOA->ODR &= ~(1U <<4);
      GPIOA->ODR &= ~(1U <<5);
}
void System_D3(){
      GPIOA->ODR |=(1<<4) ; //PA4   (16)
      GPIOA->ODR &= ~(1U <<1);
      GPIOA->ODR &= ~(1U <<0);
      GPIOA->ODR &= ~(1U <<5);
}
void System_D4(){
      GPIOA->ODR |=(1<<5) ; //PA5    (32)
      GPIOA->ODR &= ~(1U <<1);
      GPIOA->ODR &= ~(1U <<4);
      GPIOA->ODR &= ~(1U <<0);
}
void SSD_Reset(){
      GPIOA->ODR &= ~(1U <<0);
      GPIOA->ODR &= ~(1U <<1);
      GPIOA->ODR &= ~(1U <<4);
      GPIOA->ODR &= ~(1U <<5);
}
void System_Decimal_Set(){
      GPIOA->ODR |= (1<<11) ; // pa11 open the dat.
}
void System_Decimal_Reset(){
      GPIOA->ODR &=  ~(1<<11) ; // pa11  // close the dat .
}


void Delay(volatile uint32_t s) {
    for(; s>0; s--);
}
```

4 .Checkmark

- ✓ Written in C. No HAL or equivalent libraries.
- ✓ A keypad and a seven-segment display should be attached.
- ✓ On power up SSI) should show your ID (first 2 and last 2 digits).
  - As soon as a number is pressed, everything should be cleared and only your number should be displayed.
- ✓ If no button is pressed for 16 seconds, the SSD should turnoff - go back to the IDLE state.
- ✓ When keys are entered, the SSD should shift the numbers to the left, while not displaying anything for empty digits.
- ✓ If the digits are already full, new number key presses should be ignored.
- ✓ ABCDEF keys should be used as: o A is for addition o B is for subtraction o Cis for multiplication o Disfor division o E key is scientific mode, and will expect another keypress.
- ✓ EA is for log
- ✓ EB is for In
- ✓ EC is for sqrt
- ✓ ED is for $x^A 2$
- ✓ EE is for trigonometric mode, and will expect another key-press.
- ✓ EEA is for sin
- ✓ EEB is for cos
- ✓ EEC is for tan
- ✓ EED is for cot
- ✓ Scientific and trigonometric modes will require floating point number system.
- ✓ Floating point numbers should be displayed with the appropriate dot. For example if you want to show 1 .2345152 - SSD should display 1 .234 and if you want to display 4213 .123 it should display 4213.
- ✓ Negative numbers should have a negative sign. i.e -124 on the SSD.
- ✓ If thenumbers overflows 9999 or -999, it should display overflow (i.e. OuFL)
- ✓ If the operation is invalid (i.e. 3/0 or sqrt(-2) it should display invalid (i.e. Invd)
- ✓ If nokeys are pressed for 16 seconds, the SSD should turnoff. - go back to IDLE state.
- ✓ If directly a function is invoked, the current value should be used. For example, if the last answer is 4 and _ 4 is pressed, it should do 4 4 operation and display e. If in the beginning, the number should be assumed 9.

- ✓ EEE is for pi (will replace the number with 3 .141) o F key is for enter/equal

✓ That's means done.

● Not Done.

5. Missing Parts:

*There isn't any missing parts.

6. Challenges:

*It is very long Project. there is a lot to check, for example Number is positive or negative, What is operation type? (vb)

* There is very important thing that I learnd, In the begining of the Project ,when I push the button ,Keypad don't work. I have to press button a lot and then Keypad work, I understood that when I change GPIOA's ODR registers for display number, I also change Keypad R2,R1,R3,R4'values so Keypad didn't work.

*Displaying floating number also diffucult. It is not too much but it is too long.

*Bouncing phenomena is difficult, In order to prevent bouncing phenomena, a delay for a certain time was put when the interrupt came. Also all input pins was setted as puldown.

7. Conclusion:

* In this Project, how to 4x4 Keypad is controlled was learned.

* How to SSD and Keypad work simultaneous was learned. And also Learned what needs attention. For example if we change SSD's digits (a,b,c,d,e,f,g),we shouln't change all ODR register because, we are also working with keypad.

*There are a lot of requirements so when codes was written, I should write step by step, I solve problems step by step.

* Pi values was done last day.So code change last day.There are codes for pi values**.These codes normally should be in Process_Result () function that in process.c file but I can't add codes part at the report so I add below.**

```
for(int l=0;l<operation_place;l++){// if user push the EEE thats means is pi

      if(number.array_number[l]==15&&number.array_number[i+1]==15&&number.array_
number[l+2]==15)
                        number_number[0]=3.14;
            }



for(int l=operation_place;l<10;l++){// if user push the EEE thats means is pi
(this is for second number)

      if(number.array_number[l]==15&&number.array_number[i+1]==15&&number.array_
number[l+2]==15)
                              number_number[1]=3.14;
                  }
```

8.References:

* https://components101.com/misc/4x4-keypad-module-pinout-configuration-features-datasheet

* https://theokelo.co.ke/how-to-get-your-hs420361k-32-4-digit-7-segment-display-working-with-an-arduino/

* ARMv6-M Architecture Reference Manual

* dm00371828-stm32g0x1-advanced-armbased-32bit-mcus-stmicroelectronics

There are video links for projects. There 6 videos . First video is code explanation,second is debug videos . Other videos were taken to show all the features of the calculator.

(Code Explanation)

* https://drive.google.com/file/d/1dPi0hOwfRDr46aAK5vbfihtpn9zNQkTU/view?usp=sharing

(Debug)

* https://drive.google.com/file/d/1fr1xyvoTwdc61P8wiJOSyzf54YKi8v-j/view?usp=sharing

(Operation with Negative Numbers and Overflow (x>9999))

* https://drive.google.com/file/d/1z2x6I-pB1ODzmGcF3Um2r4wpUoTw6ssz/view?usp=sharing

(Sqrt and Sqrt’s overflow)

* https://drive.google.com/file/d/1mck5dgo-9OnlGPOjGascNVTGnInfP17q/view?usp=sharing

(Trigonometric Operations)

* https://drive.google.com/file/d/1RVqvhXDfLw_DaB9wJ9ryf7NVVchC4t2D/view?usp=sharing

(Pi Values)

*https://drive.google.com/file/d/1WYjxTnxlib_B8RfD64MwyLzzIW_FQs0m/view?usp=sharing