

```

/*****
* Name:      ADC.c
* Description: STM32 Analogy to Digital conversion
* Version: V1.00
* Authors: Li Pan
*
*****/
#include "stm32f10x.h"
#include "ADC.h"
#include "LCD.h"
#include "CLOCK.h"

// ADC enable, PinA enable
void ADC_INIT(void)

{
    RCC->APB2ENR |=RCC_APB2ENR_ADC1EN | RCC_APB2ENR_AFIOEN |
RCC_APB2ENR_IOPAEN;

    GPIOA->CRL &= 0x44440444;

    ADC1->CR2 |=0x1;

}
/*read different channel for output */
uint32_t ADC_Read(channel)
{
    ADC1->SQR3=channel;

    ADC1->CR2 |= 0x0000001;

    while((ADC1->SR & ADC_SR_EOC) != ADC_SR_EOC)
    {

    }
    return ADC1->DR;
}
/*hex value display*/
void Hex2LCD(void)
{
    int channel=1;
    int i;
    uint32_t temp;

    while(1)
    {
        if(channel==1)
        {
            //STR2LCD("LM35:");
            temp = ADC_Read(channel);

            for(i=28;i>=0;i=i-4)

```

```

        {

            DATA2LCD(Hex2Ascii((temp>>i) &0xF));

        }

        STR2LCD("-");
        Float2LCD((temp*0.08),channel);

    }

    CMD2LCD(LCD_LN2);

    channel++;

    if (channel==2)
    {

        temp=ADC_Read(channel);

        for(i=28;i>=0;i=i-4)
        {

            DATA2LCD(Hex2Ascii((temp>>i) &0xF));

        }

        STR2LCD("-");
        Float2LCD((0.8*temp),channel);

    }

}

//check the ADC inout for LCD display the different drive condition
void BRIGHT2LCD(void)
{

    int channel=3;

    uint32_t temp = ADC_Read(channel);

    CMD2LCD(LCD_LN1);

    RCC->APB2ENR |= RCC_APB2ENR_IOPBEN;
    GPIOB->CRH |= GPIO_CRH_MODE10 |GPIO_CRH_MODE11;
    GPIOB->CRH &= ~GPIO_CRH_CNF10 &~GPIO_CRH_CNF11;

    //initial the biggest value for LCD display

```

```

if (temp>0x00000900)
{
    STR2LCD("Safe      ");

    GPIOB->BSRR = GPIO_BSRR_BR10;
    GPIOB->BSRR = GPIO_BSRR_BR11;

}
//if the input value is lower , turn on light
else if(temp>0x780)
{
    STR2LCD("Dark      ");
    GPIOB->BSRR = GPIO_BSRR_BS10;
    GPIOB->BSRR = GPIO_BSRR_BR11;
    delay(60000);
}
//if the inout value is lower than preview value, turn on all
driving lights
else
{
    STR2LCD("Very Dark");
    GPIOB->BSRR = GPIO_BSRR_BS10;
    GPIOB->BSRR = GPIO_BSRR_BS11;
    delay(60000);
}

}

```