

```
/* Name: David (DongYun) Kim
 * SID: 200405213
 * Course: ENEL351
 * Description: ENEL351 Project - Smart Parking System
 * File name: adc.c
 */
```

```
#include "stm32f10x.h"
#include "adc.h"
```

```
void adc1_init(void) {
    // Enable the clocks for ADC1, port A, AFIO
    RCC -> APB2ENR |= (RCC_APB2ENR_ADC1EN | RCC_APB2ENR_IOPAEN | RCC_APB2ENR_AFIOEN);
    RCC -> CFGR |= RCC_CFGR_ADCPRE; // divide APB2 by 8 (72/8 = 9Mhz)

    // Configure PA0 as an analog input (MODE/CNF = 0000)
    GPIOA -> CRL &= ~GPIO_CRL_MODE0;
    GPIOA -> CRL &= ~GPIO_CRL_CNF0;

    // Turn on ADC1
    ADC1 -> CR2 |= ADC_CR2_ADON;

    ADC1 -> SMPR2 |= ADC_SMPR2_SMP0; // Set all bits to 1 for the slowest sampling time for PA0

    //Turn on ADC1
    //ADC1->CR2 = 0x00000001;
    // Run an ADC calibration
    ADC1 -> CR2 |= ADC_CR2_CAL; // Start calibration
    while ((ADC1 -> CR2 & ADC_CR2_CAL) == ADC_CR2_CAL); // Wait for calibration to finish
}

int adc1_acquire(void)
{
    // Averaging 10 readings for increased stability
    int sum = 0;
    for(int i = 0; i < 10; i++)
    {
        // Select PA0 as the channel to convert (Channel 0)
        ADC1->SQR3 = 0; // ADC1->SQR3 bits [4:0] are for the 1st conversion in regular
        sequence, set to 0 for channel 0 (PA0)

        // Start the conversion
        ADC1->CR2 |= ADC_CR2_ADON; // Setting ADON bit starts the conversion

        // Wait for the conversion to complete
        while (!(ADC1->SR & ADC_SR_EOC)); // Wait for EOC (End of conversion) flag

        sum += ADC1->DR & ADC_DR_DATA; // Read conversion result and accumulate sum
    }
    return sum / 10; // Return the average value
}
```