

WE ARE USING STM32F373VCT6 LQFP IC

WE HAVE SIX ANALOG INPUTS AT PIN NUMBERS

38 - SDADC1-AIN3P ----- INPUT1

39 - SDADC1-AIN8P ----- INPUT2

40 - SDADC1-AIN7P ----- INPUT3

41 - SDADC1-AIN2P ----- INPUT4

42 - SDADC1-AIN1P ----- INPUT5

43 - SDSDC1-AIN0P ----- INPUT6

VREF =1.8V INTERNAL

AS WE HAVE OBSERVED CROSS TALK BETWEEN THE CHANNELS

IF WE APPLY CONSTANT DC VOLTAGE TO THE ONE INPUT SAY INPUT1 .

AND NOW WE VARY INPUT VOLTAGE AT ANY OTHER INPUTS THEN COUNTS AT ?INPUTS1? GET CHANGED. WHILE INPUT VOLTAGE AT THIS PIN REMAIN SAME .

I HAVE TAKEN SOME READINGS.

VREF =INTERNAL 1.8V

PGA CONFIGURED =1 ;

AIN VOLT COUNT AT INPUT1

INPUT1 0.752V 28008

INPUT2 170mV

INPUT3 190mV

INPUT1 0.752V 27982

INPUT2 170mV

INPUT3 190mV

INPUT1 0.752V 27996

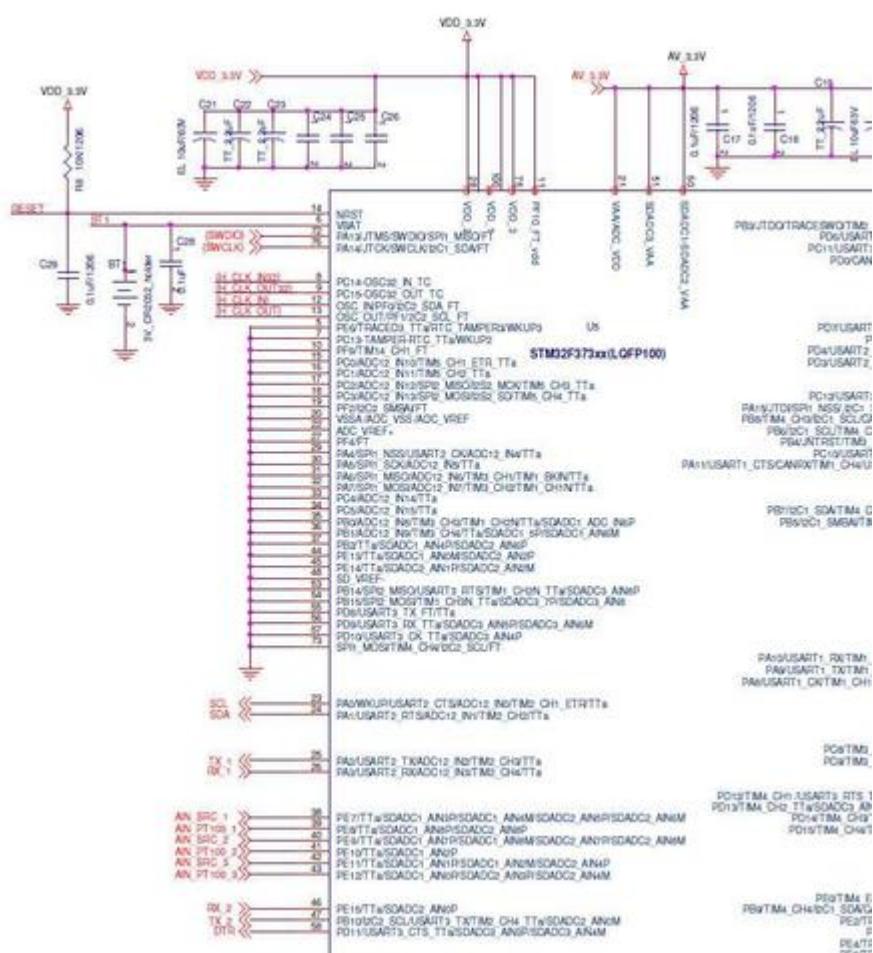
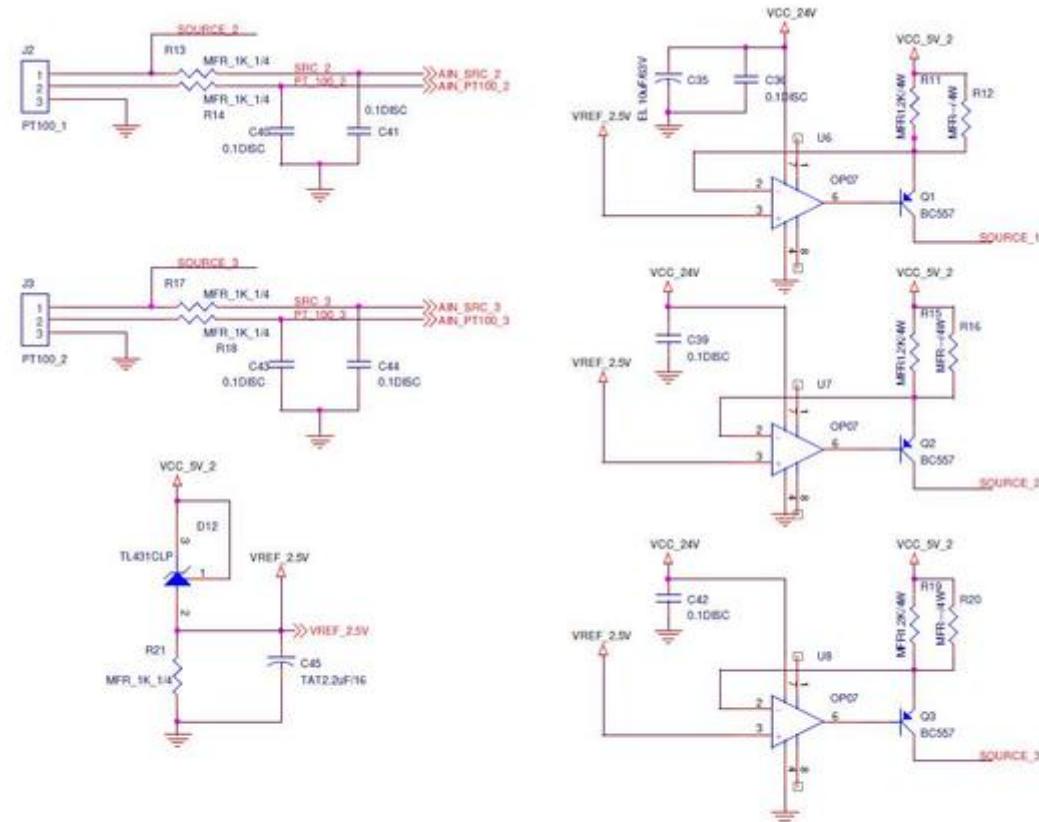
INPUT2 170mV

INPUT3 190mV

ALSO I HAVE OBSERVED THAT IF I CONFIGURE ONLY ONE INPUT 1 AS ANALOG INPUT REMAINING INPUTS CONFIGURED AS OPEN DRAIN DIGITAL INPUTS . NOW ONLY SINGLE ADC CHANNEL IS THERE .BUT HERE ALSO WHEN I VARY VOLTAGE ? 0V? TO ? 1.8V? ON OTHER INPUTS(WHICH ARE CONFIGURED AS DIGITAL INPUTS) THEN ALSO COUNTS ON ADC INPUT1 CHANGES.

PLEASE GUIDE ME TO OVERCOME THIS PROBLEM.

I HAVE ALSO ATTACHED ANALOG CIRCUIT WHICH WE ARE USING .



Here id code for sdadc

```
#include "stm32f37x.h"
```

```
#define SDADC1_INJ_DR_ADDRS 0x40016070
#define PT100_CH1 SDADC_Channel_8
```

```
#define PT100_CH1_PIN GPIO_Pin_8
#define PT100_CH1_PORT GPIOE
```

```
#define PT100_CH2_PIN GPIO_Pin_7
#define PT100_CH2_PORT GPIOE
```

```
#define PT100_CH3_PIN GPIO_Pin_9
#define PT100_CH3_PORT GPIOE
```

```
#define PT100_CH4_PIN GPIO_Pin_10
#define PT100_CH4_PORT GPIOE
```

```
#define PT100_CH5_PIN GPIO_Pin_11
#define PT100_CH5_PORT GPIOE
```

```
#define PT100_CH6_PIN GPIO_Pin_12
#define PT100_CH6_PORT GPIOE
```

```
GPIO_InitTypeDef GPIO_InitStructure;
```

```
DMA_InitTypeDef DMA_InitStructure ;
```

```
SDADC_InitTypeDef SDADC_InitStructure;
```

```
SDADCAINStructTypeDef SDADC_AINStructure;
```

```
NVIC_InitTypeDef NVIC_InitStructure;
```

```
///////////////////////////////
```

```
vu32 Inje_Dma_Data[20],average_Data;
```

```
vu8 adc_ready,rd_num ;
```

```
vu16 Adc_Data[20],Final_reading;
```

```
vs16 Inj_Conv_Data[20] ;
```

```
///////////////////////////////
```

```
void SDADC_Config(void);
```

```
void TIM19_Config(void); //EXTERNAL TRIGGER FOR SDADC
```

```
void DMA2_CONFIG(void) ;
```

```
void Delay_SW(vu32 nCount);
```

```
void cal_avrg(void) ;
```

```

void Delay_SW(vu32 nCount)
{
    for(; nCount!= 0;nCount--);
}

void main(void)
{
//rcc is configured for 72 Mhz
    TIM19_Config();
    DMA2_CONFIG();
    SDADC_Config();
    while(1)
    {
        if(adc_ready==1) //this bit is set in Dma Transfer complete interrupt
        {
            adc_ready=0 ;
            cal_avrg();
        }
    }
}
///////////
void cal_avrg(void)
{
    average_Data=0;
    rd_num=0 ;
    for(rd_num=0;rd_num<20;rd_num++)
    {
        Inj_Conv_Data[rd_num]=(vs16)(Inje_Dma_Data[rd_num]);

        Adc_Data[rd_num] = Inj_Conv_Data[rd_num] + 32768;

        average_Data=average_Data+Adc_Data[rd_num];
    }
    average_Data=average_Data/20 ;

    Final_reading=average_Data ; //Final adc reading .
}

///////////
void SDADC_Config(void)
{
    uint32_t SDADCTimeout = 0;
    SDADC_DeInit(SDADC1);

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_SDADC1, ENABLE);

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR, ENABLE);

    PWR_SDADCAnalogCmd(PWR_SDADCAnalog_1, ENABLE);

    RCC_SDADCCLKConfig(RCC_SDADCCLK_SYSCLK_Div48);

    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOE, ENABLE);
}

```

//only one channel is configuerd as adc input all other five input ihave configured as

```

//open drain digital input . for checking .
GPIO_InitStructure.GPIO_Pin = PT100_CH1_PIN;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AN;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_Init(PT100_CH1_PORT, &GPIO_InitStructure);

SDADC_VREFSelect(SDADC_VREF_VREFINT2);

Delay_SW(1000);

SDADC_Cmd(SDADC1, ENABLE);

SDADC_InitModeCmd(SDADC1, ENABLE);

SDADCTimeout=200 ;
while((SDADC_GetFlagStatus(SDADC1, SDADC_FLAG_INITRDY) == RESET) && (--SDADCTimeout != 0));
Delay_SW(1000);

SDADC_Cmd(SDADC1, DISABLE);

SDADC_AINStructure.SDADC_InputMode = SDADC_InputMode_SEZeroReference;
SDADC_AINStructure.SDADC_Gain = SDADC_Gain_4;
SDADC_AINStructure.SDADC_CommonMode = SDADC_CommonMode_VSSA;
SDADC_AINStructure.SDADC_Offset = 0;
SDADC_AINInit(SDADC1, SDADC_Conf_0, &SDADC_AINStructure);

SDADC_Cmd(SDADC1, ENABLE);

SDADC_ChannelConfig(SDADC1, PT100_CH1, SDADC_Conf_0);// SDADC_Channel_8
SDADC_InjectedChannelSelect(SDADC1, (PT100_CH1));
SDADC_ExternalTrigInjectedConvConfig(SDADC1, SDADC_ExternalTrigInjecConv_T19_CC2);
SDADC_ExternalTrigInjectedConvEdgeConfig(SDADC1, SDADC_ExternalTrigInjecConvEdge_Rising);
SDADC_InitModeCmd(SDADC1, DISABLE);

}

void TIM19_Config(void)
{
    TIM_OCInitTypeDef TIM_OCInitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;

    /* Enable TIM19 clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM19, ENABLE);

    /* TIM19 Configuration */
    TIM_DeInit(TIM19);
}

```

```
/* Fills each TIM_TimeBaseInitStruct member with its default value */
TIM_TimeBaseStructInit(&TIM_TimeBaseStructure);

/* Time base configuration: MPX2102_SDADC will be triggered each sysclk/Period
 = 72MHz/20000 = 3.6 KHz */
TIM_TimeBaseStructure.TIM_Period = 20000 ;//277 micro sec per sample
TIM_TimeBaseStructure.TIM_Prescaler = 0 ;
TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM19, &TIM_TimeBaseStructure);
```

```
/* PWM1 Mode configuration: Channel2 (OC2) */
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = 20000;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OC2Init(TIM19, &TIM_OCInitStructure);
```

```
/* Enable TIM19 counter */
TIM_Cmd(TIM19, ENABLE);
}
/////////
```

```
void DMA2_CONFIG(void)
{
/* DMA1 clock enable */
RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA2 , ENABLE);
```

```
/* DMA2 Channel3 Config */
DMA_DeInit(DMA2_Channel4);
DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)SDADC1_INJ_DR_ADDRS;
DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)Inje_Dma_Data;
DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
DMA_InitStructure.DMA_BufferSize = 20;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_Word;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_Word;
DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
DMA_InitStructure.DMA_Priority = DMA_Priority_High;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA2_Channel4, &DMA_InitStructure);
```

```
// Enable the DMA1 Channel1 Interrupt
NVIC_InitStructure.NVIC_IRQChannel = DMA2_Channel4 IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 4;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
```

```
NVIC_Init(&NVIC_InitStructure);
```

```
/* DMA1 Channel1 enable */  
DMA_Cmd(DMA2_Channel4, ENABLE);  
DMA_ITConfig(DMA2_Channel4, DMA_IT_TC, DISABLE);
```

```
}
```