基于STM32G474和Simulink的信号发生器 & 示 波器

实现功能

实现的功能具体而言有两部分,分别是信号发生器和示波器。

波形发生器实现的功能:

最终实现了能够实现正弦波、方波、三角波的波形发生器,产生频率为20Hz,幅值为3.3V。按下B1键可以切换模式,切换的顺序为正弦波-->方波-->三角波-->正弦波...

示波器实现的功能:

通过simulink的display显示出信号发生器波形的图样。

演示视频:

在simulink上观察波形:

https://cloud.tsinghua.edu.cn/f/fc6f3cb4e59e48d7a529/

在matlab上观察波形:

https://cloud.tsinghua.edu.cn/f/1dd7012ca2784243a59b/

实现方法

波形发生器

基于波形的数据信息和单片机的DAC (DMA) 模块产生波形信号,由于本项目我们有三个波形, 所以我们需要三组数据信息,我们使用MATLAB产生波形信息。

```
plot(SineData);
Fid = fopen('HarmonicSineWaveData.txt','w');
fprintf(Fid, '%d, ', SineData);
fclose(Fid);
A=4096/2-1;%信号幅值
N=200:%-- 周期内数据占数
Ph=0;%初始相位
\label{eq:sineData} SineData=ceil(A*sin(Ph:2*pi/N:2*pi*(1-1/N)+Ph)+A);
plot(SineData):
Fid = fopen('SineWave1Data.txt','w');
fprintf(Fid,'%d,',SineData);
                                                                                                                              3500
fclose(Fid);
                                                                                                                              3000
                                                                                                                              2500
% 方波
A=4096/2-1;%信号幅值
N=200;%一周期内数据点数
Ph=0;%初始相位
SquareData=[A*2 *ones(1, N/2) zeros(1, N/2)];
plot(SquareData);
Fid = fopen('Squa
                 areWaveData.txt','w');
                                                                                                                            riangleData = 1×200 int16 row v
fprintf(Fid,'%d,',SquareData);
                                                                                                                                2047 2089 2131 2172 2214 ...
fclose(Fid);
% 三角波
A=4096/2-1;%信号幅值
N=200;%一周期内数据点数
Ph=0;%初始相位
riangleData=int16([linspace(0, A, N/4) linspace(A, 0, N/4) linspace(0, -A, N/4) linspace(-A, 0, N/4)] + A)
plot(TriangleData);
Fid = fopen('TriangleWaveData.txt','w');
fprintf(Fid,'%d,',TriangleData);
fclose(Fid);
```

得到波形数据后,我们存入数组中。

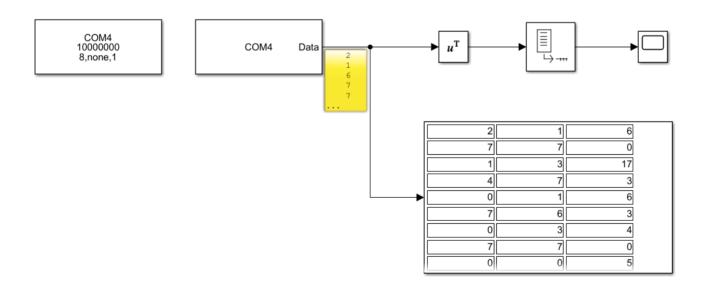
为了切换了不同的模式,我们采用按键(B1)中断的方式,按下按键之后,触发中断,在中断中进行模式切换。

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    HAL_Delay(10);
    if (GPIO_Pin == KEY_Pin){
        mode = (mode + 1) % MODE_NUM;
        HAL_DAC_Stop_DMA(&hdac1, DAC1_CHANNEL_1);
        switch(mode){
             case SINE_MODE:{
                 HAL_DAC_Start_DMA(&hdac1, DAC_CHANNEL_1, (uint32 t *)Sine1WaveData, DAC_BUFFER_SIZE, DAC_ALIGN_12B_R);
                 break;
             }
             case SQUARE_MODE:{
                 HAL_DAC_Start_DMA(&hdac1, DAC_CHANNEL_1,(uint32_t *)SquareWaveData,DAC_BUFFER_SIZE,DAC_ALIGN_12B_R);
                  break;
             case TRIANGLE MODE:{
                 HAL_DAC_Start_DMA(&hdac1, DAC_CHANNEL_1,(uint32_t *)TriangleWaveData,DAC_BUFFER_SIZE,DAC_ALIGN_12B_R);
             }
        }
    }
}
```

示波器

示波器的实现是通过STM32的DAC采样信号发生器的信号,再通过串口发送给PC机,PC机上的 Simulink Model将串口的数据绘制成波形。

Simulink模型:



串口发送数据:

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* AdcHandle)
{
    HAL_UART_Transmit(&huart2, (uint8_t *)&FrameHeader,2, 0xFFFF);
    HAL_UART_Transmit(&huart2, (uint8_t *)&ADC1ConvertedData,
    ADC_CONVERTED_DATA_BUFFER_SIZE*2, 0xFFFF);
    HAL_UART_Transmit(&huart2, (uint8_t *)&FrameTerm,2, 0xFFFF);
}
```

小结

通过嵌入式系统实践这门课,我们了解到了STM32单片机的内部结构,GPIO、中断、串口、时钟 (PWM)、DAC-ADC、DMA等功能,并通过一些实践项目学习了相关的编程技巧和调试经验, 收获匪浅,感谢王鹏老师和朱小梅老师的指导!!