

HDMI 波形显示例程

1、实验简介

在回环例程的基础上，我们将 ADC 回环采集回来的波形，通过 HDMI 显示在显示器上。实验中使用“小眼睛科技”的 MES-ADDA-v1.0 模块，该模块配有杭州瑞盟公司的 MS9708 数模转换芯片与 MS9280 模数转换芯片，因此同时具有数模转换与模数转换的功能。

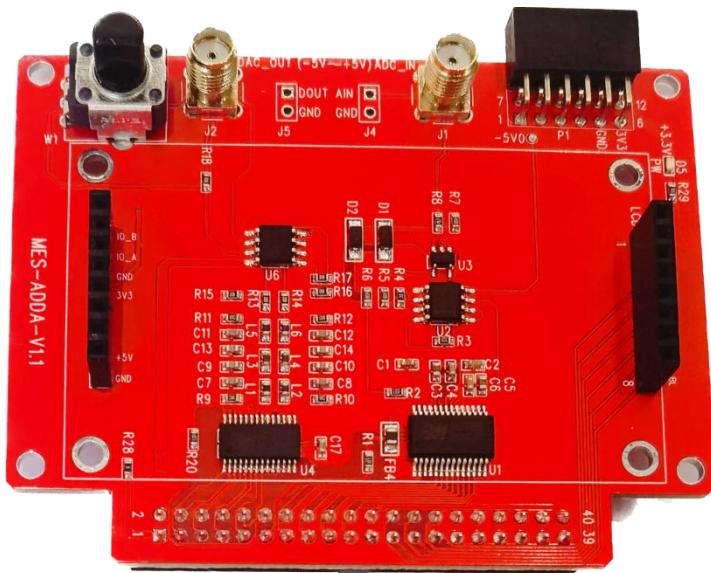


图 1 MES-ADDA-v1.0 模块

实验器件准备：

FPGA 开发板、MES-ADDA-v1.0 模块、显示器。

实验流程如下：

- 1、对 DA 数模转换芯片的模拟输出与 AD 模拟芯片的模拟输入进行连接。
- 2、将显示器与 FPGA 开发板进行连接。
- 3、代码的烧录。
- 4、观察显示器上显示的波形是否符合预期。

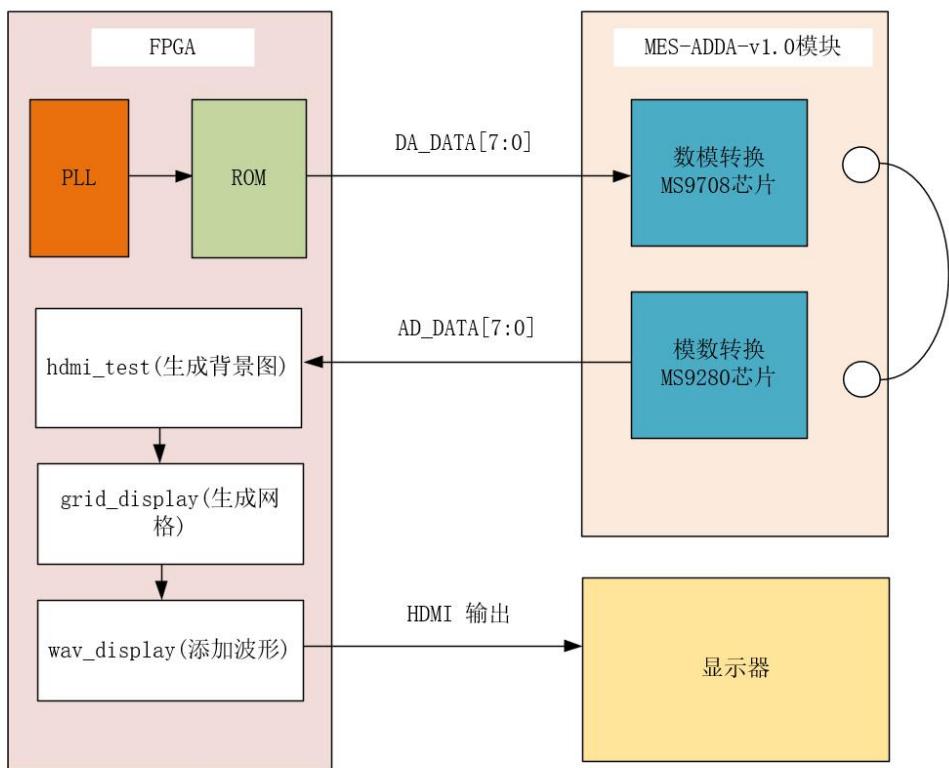


图 2 实验流程图

芯片参数如下：

MS9708 芯片参数：

- 8bit 分辨率(MS9708), 10bit 分辨率(MS9710), 14bit 分辨率(MS9714)
- 更新速率：125MSPS
- 功耗：175mW @ 5V 到 45mW @ 3V
- 掉电模式：20mW @ 5V
- 内部基准：1.2V
- 边沿触发锁存器
- TSSOP28 封装

MS9280 芯片参数

- 10 bit、35 MSPS 流水线 ADC (只使用高 8bit 数据输入)
- 低功耗：90mV (3V 电源下)
- 宽工作范围：2.7V ~ 5.5V
- 高线性度: DNL : 0.2 LSB
- 低功耗控制模式
- 三态门输出
- 量化范围检测
- 内建钳位功能
- 高精度可编程基准电源
- 中频亚采样高达

2、实验原理

FPGA 通过 40pin 扩展口将数字信号输出给 DAC 芯片，经过低通滤波器、幅度调节，输出模拟信号，再将输出的模拟信号接回到 MES-ADDA-v1.0 模块的 ADC 模拟输入接口，经过衰减电路，ADC 芯片，将 ADC 采集到的数据输入到 FPGA 上，使用显示器显示 ADC 采集到的数据。

2.1 DAC 输出模拟信号

模数转换部分的流程图如下：

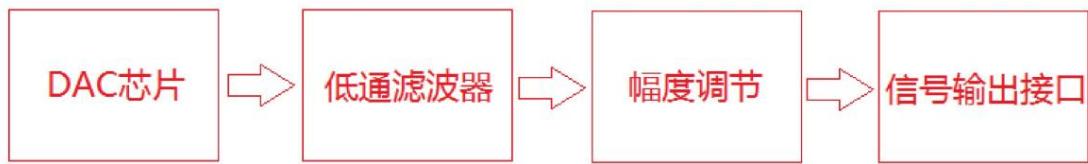


图 3 数模转换流程

DAC 芯片将数据进行数模转换，输出电流信号，转换关系如下：

$$I_{OUTA} = (\text{DAC CODE}/256) \times I_{OUTFS} \quad (1)$$

$$I_{OUTB} = (255 - \text{DAC CODE})/256 \times I_{OUTFS} \quad (2)$$

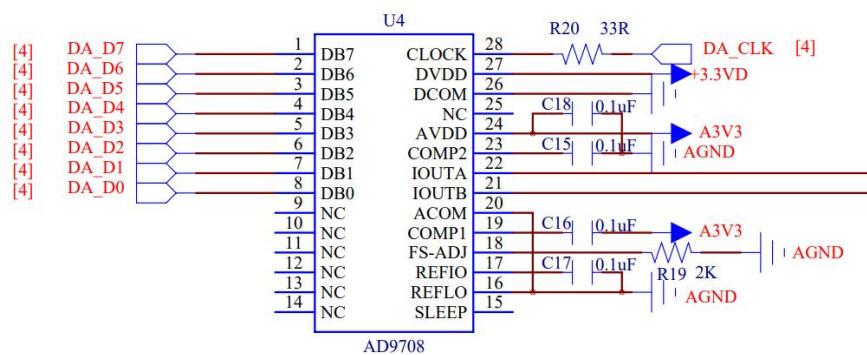
$$I_{OUTFS} = 32 \times I_{REF} \quad (3)$$

$$I_{REF} = V_{REFIO} / R_{SET} \quad (4)$$

注：

1、RSET 为图中 R19，REFIO 接地，故使用的是内部 1.2V 基准，VREFIO 为 1.2V。

2、电阻 RSET 在原理图中为 R19。



低通滤波电路用于滤波和去噪，使输出波形更加圆滑，如下图所示：

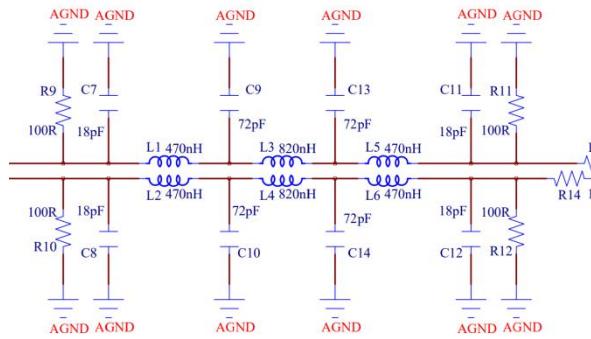


图 5 低通滤波电路

第一级负反馈放大电路的目的使将输出电流信号转换为电压信号, 第一级负反馈放大电路的输出电压范围为 : -1v ~ +1v。

电路如下图所示 :

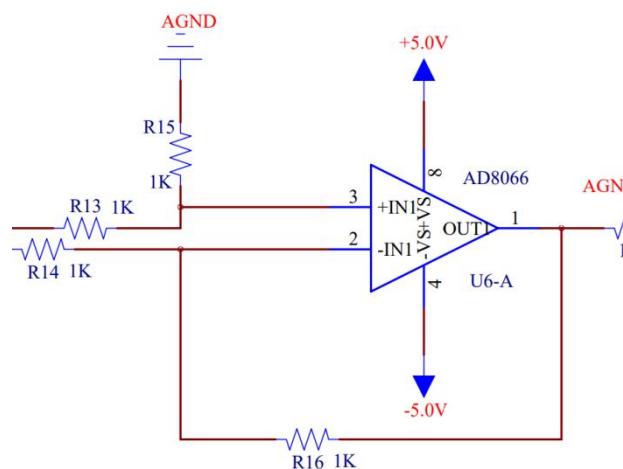


图 6 第一级放大电路

第二级负反馈放大电路通过对 W1 旋钮电阻的调节, 控制放大倍数, 但是如果调剂不当, 会导致波形失真。测试显示第二级放大电路的输出电压范围在 +3.3V ~ -3.3V 内时, 输出波形不失真。

设 W1 的阻值为 R_f , 输出电压 V_{out} , 输入电压 V_{in} , 它们的关系如下 :

$$V_{out} = - (R_f/R_{17}) * V_{in} ;$$

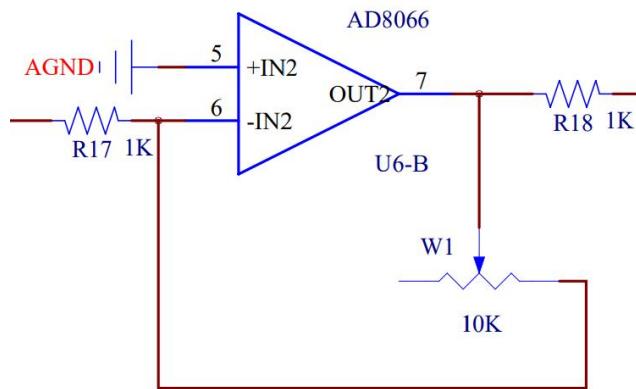


图 7 第二级负反馈放大电路

输出端 IOUTB 为互补 DAC 电流输出端，经过两次负反馈放大电路后，波形仍然为互补的形式，所以输出的波形与原始数字信号输入波形相比，是相反的。

2.2 ADC 输入模拟信号

流程图如下：



图 8 模数转换流程

模拟信号输入后，需要根据 ADC 芯片管脚的电气特性将数据进行衰减，衰减的部分电路如下图所示：

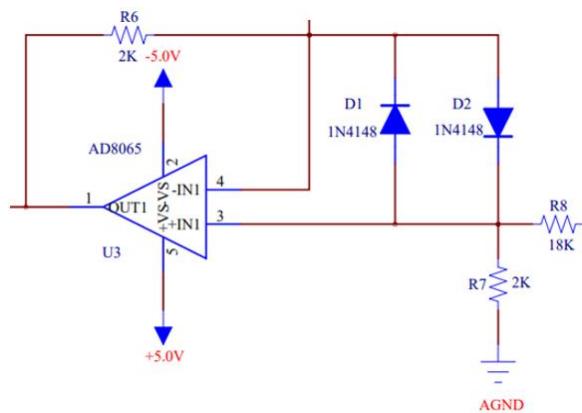


图 9 衰减电路

ADC 芯片用于将模拟信号转换为数字信号，电路图示如下图所示：

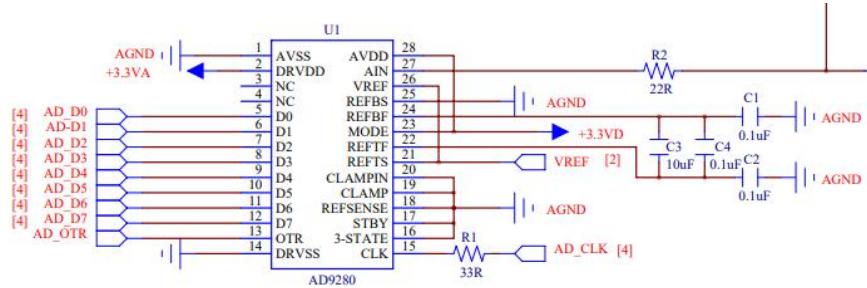


图 10 ADC 芯片

3、程序设计

3.1 实验代码

例程中，输入时钟的频率为 50MHz，使用 PLL 生成 DAC 时钟频率与 ADC 时钟频率，如图，我们将 DAC 的时钟频率设置为了 118.8MHz，ADC 的时钟频率设置为了 29.7MHz。可以根据需要调整 DAC 与 ADC 时钟频率，但因注意，DAC 时钟频率不应大于 125MHz，ADC 时钟频率不应大于 35MHz。使用 matlab 生成 ROM 初始化文件.dat 在正弦波例程中已经说明过，这里不再过多赘述。

```

1  module ad_da_hdmi_top(
2    input  wire  clk_50M      ,
3    input  wire  rst_n       ,
4    output wire  rstn_out    ,
5    //HDMI config
6    output wire  iic_tx_scl  ,
7    inout   wire  iic_tx_sda  ,
8    output wire  led_int     ,
9    //ADDA
10   input  wire  [7:0]ad_data_in  ,
11   output wire  ad_clk      ,
12   output wire  da_clk      ,
13   output wire  [7:0]da_data_out ,
14   output wire  vout_hs     ,
15   output wire  vout_vs     ,
16   output wire  vout_de     ,
17   output wire  vout_clk    ,
18   output wire  [23:0]vout_data
19  );
20
21 //wire clk_125M ;
22 wire lock ;
23 wire pll_lock ;
24 wire clk_118_8 ;
25 wire clk_29_7 ;
26
27 wire pix_clk ;
28 wire [7:0] hdmi_r_out ;
29 wire [7:0] hdmi_g_out ;
30 wire [7:0] hdmi_b_out ;
31 wire hdmi_vs_out ;
32 wire hdmi_hs_out ;
33 wire hdmi_de_out ;
34
35 wire [23:0] grid_data_out ;
36 wire grid_vs_out ;
37 wire grid_hs_out ;
38 wire grid_de_out ;
39
40 assign vout_clk = pix_clk ;

```

例程中 hdmi_test 模块对 FPGA 板上的 HDMI 输出芯片进行配置, 并且生成背景图, grid_display 模块生成网格图, wav_display 模块用于显示波形信息。

```
47  √ pll_adda u_pll_adda (
48      .clkin1(clk_50M),           // input
49      .pll_lock(pll_lock),       // output
50      .clkout0(clk_118_8),       // output //118.8Mhz
51      .clkout1(clk_29_7)         // output //29.7Mhz
52  );
53
54  assign ad_clk = clk_29_7 ;
55
56  //output color bar
57  √ hdmi_test  hdmi_color(
58      .sys_clk      (clk_50M  ) ,// input system clock 50MHz
59      .rstn_out     (rstn_out ) ,
60      .iic_tx_scl   (iic_tx_scl) ,
61      .iic_tx_sda   (iic_tx_sda) ,
62      .led_int      (led_int  ) ,
63      .pix_clk      (pix_clk  ) ,//pixclk
64      .vs_out       (hdmi_vs_out ) ,
65      .hs_out       (hdmi_hs_out ) ,
66      .de_out       (hdmi_de_out ) ,
67      .r_out        (hdmi_r_out ) ,
68      .g_out        (hdmi_g_out ) ,
69      .b_out        (hdmi_b_out )
70  );
71
72  //output grid
73  √ grid_display grid_display_1(
74      .rst_n       (rst_n    ) ,
75      .pclk        (pix_clk  ) ,
76      .i_hs        (hdmi_hs_out) ,
77      .i_vs        (hdmi_vs_out) ,
78      .i_de        (hdmi_de_out) ,
79      .i_data      ({hdmi_r_out[7:0] , hdmi_g_out[7:0] , hdmi_b_out[7:0]}) ,
80      .o_hs        (grid_hs_out) ,
81      .o_vs        (grid_vs_out) ,
82      .o_de        (grid_de_out) ,
83      .o_data      (grid_data_out)
84  );
85
```

```

85
86  //output hdmi wave
87  `wav_display wav_display_1(
88    .rst_n      (rst_n      ) ,
89    .pclk       (pix_clk    ) ,
90    .wave_color (24'hff0000) ,
91    .ad_clk     (ad_clk     ) ,
92    .ad_data_in (ad_data_in) ,
93    .i_hs       (grid_hs_out) ,
94    .i_vs       (grid_vs_out) ,
95    .i_de       (grid_de_out) ,
96    .i_data     (grid_data_out) ,
97    .o_hs       (vout_hs    ) ,
98    .o_vs       (vout_vs    ) ,
99    .o_de       (vout_de    ) ,
100   .o_data     (vout_data  )
101 );
102
103
104 //output wave
105
106 `ad9708_sin_wave a_sin_wave (
107   .clk        (clk_118_8) ,
108   .rst_n     (rst_n) ,
109   .da_clk    (da_clk) ,
110   .da_data   (da_data_out) // 116Khz sine wave
111 );
112
113
114 endmodule

```

4、实验现象

4.1 观察实验结果前步骤

- 1、将 MES-ADDA-v1.0 模块插入到 FPGA 开发板上（40pin 扩展脚位置），如图 11。
- 2、将 MES-ADDA-v1.0 模块上 DA 模拟输出与 AD 模拟输入连接在一起，如图 12。
- 3、连接 FPGA 开发板与显示器，如图 13。
- 4、代码烧录。
- 5、调整 MES-ADDA-v1.0 模块的 W1 电阻旋钮，调整输出波形幅度，如图 14。

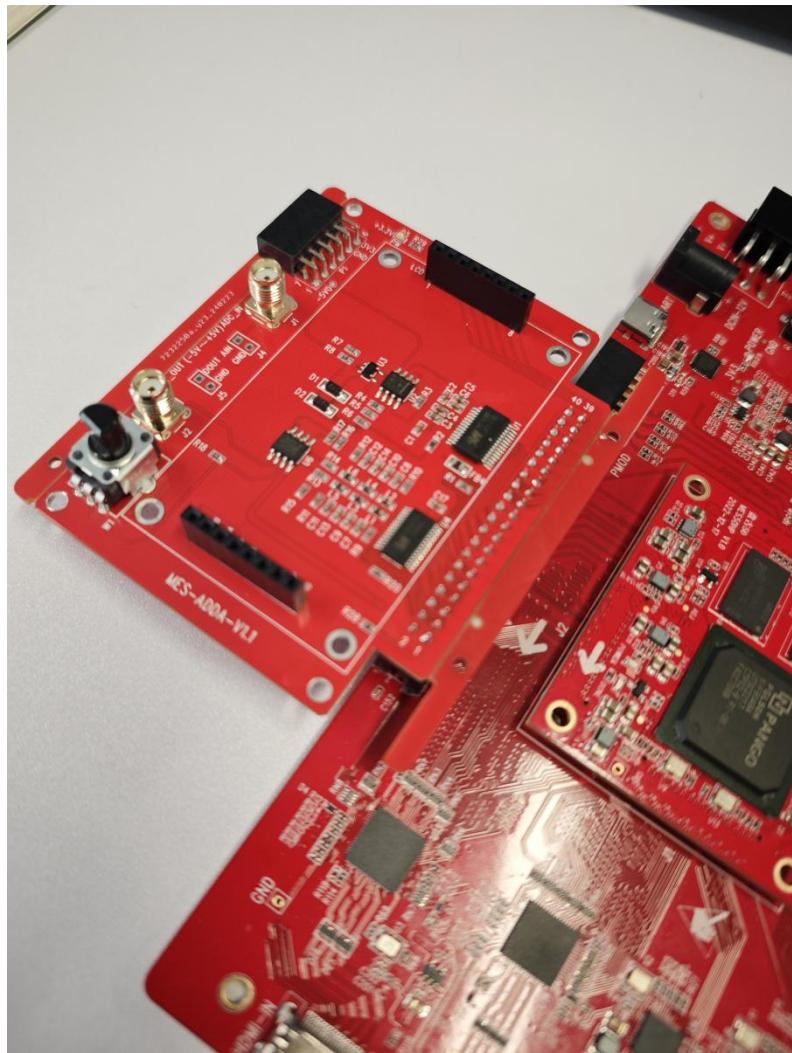


图 11 步骤一

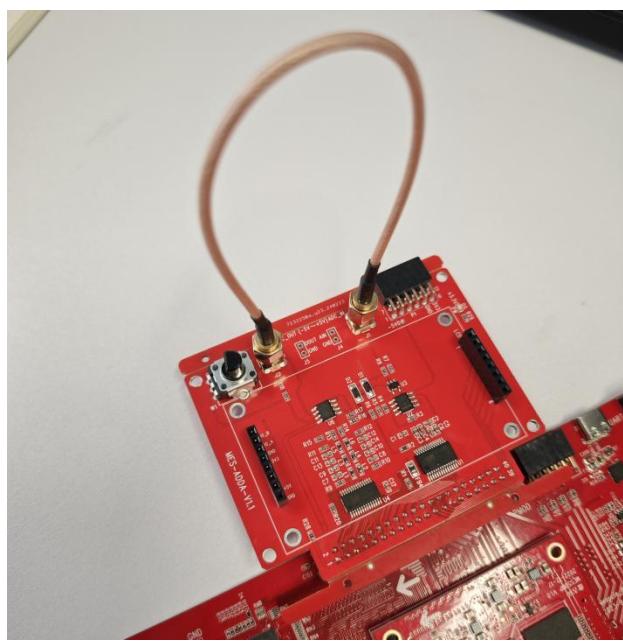


图 12 步骤二

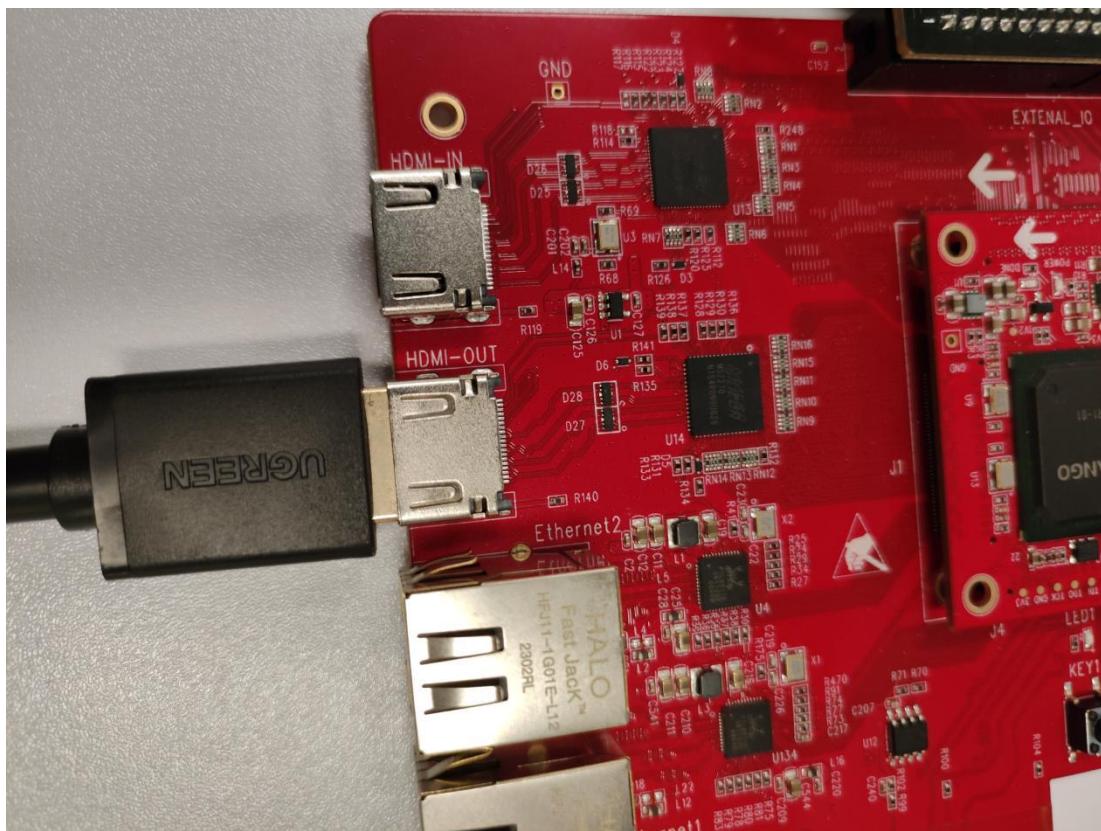


图 13 步骤三

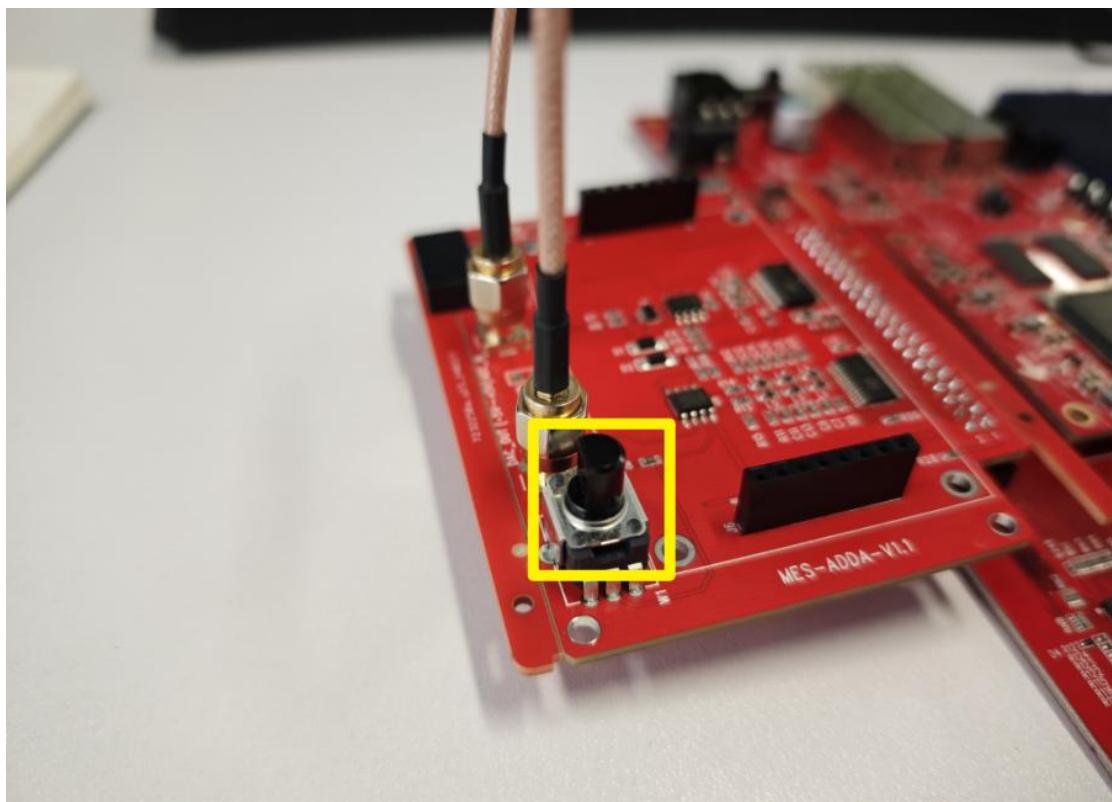


图 14 步骤五

4.2 实验现象

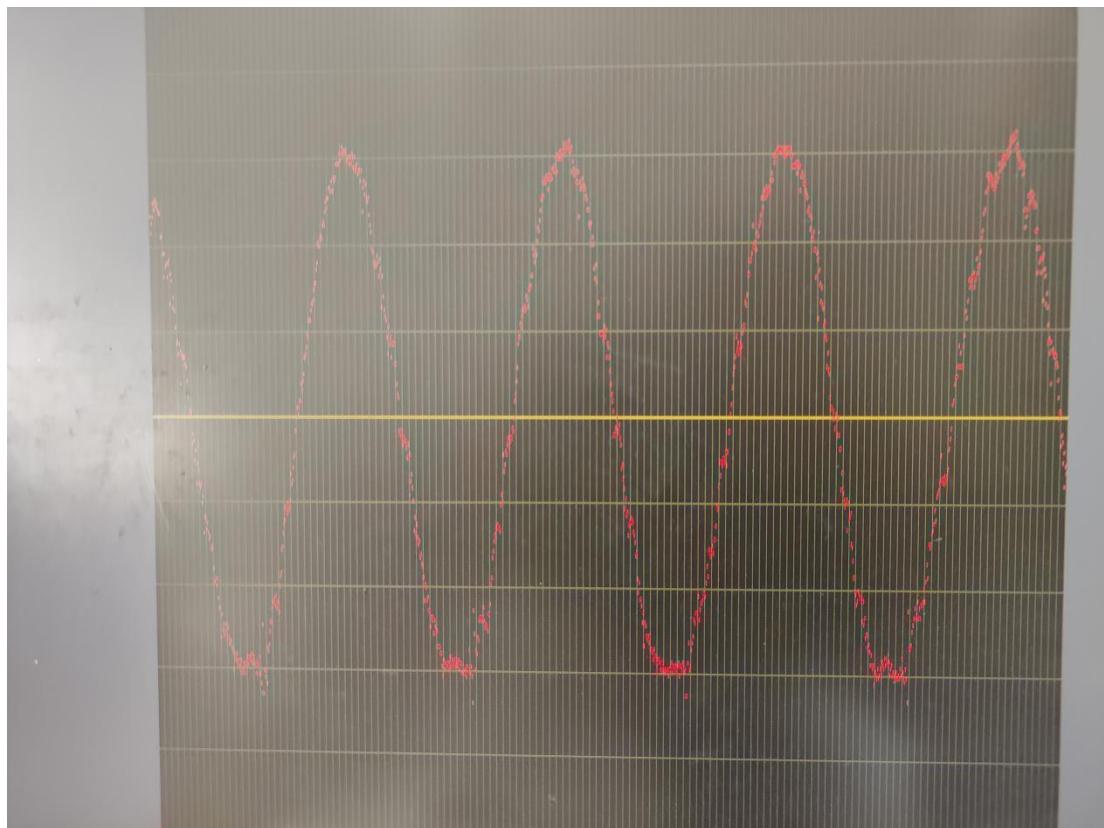


图 15 实验结果