

ACM7606 数据采集 DDR3 缓存串口发送实验


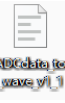
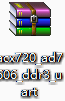

1.1. 文件说明

本文档如有更新，会发布在 www.corecourse.cn 网站。如您需查看并获取更新版本，只需要在该网站以“ACM7606”为关键词搜索，即可查找到相关内容。

版本记录：

V1.0	首次发布	2022.10.24
.....

本次实验配套的文件压缩包名为 `acx720_ad7606_ddr3_uart_V1.0.rar`，解压后可得到配套的文件工程，采样数据和调试工具等，各文件功能如下所示：

	串口猎人（Serial Hunter V31 setup）：串口调试软件，使用该软件发送启动传输、设置采集速率、设置采集通道、设置采集数量等指令，并通过该软件将采集到的数据保存为.txt 格式的文件，供后续进行数据分析。
	ADCdata_to_wave_v1_1.m：Matlab 源文件代码，对于串口传输过来的数据，我们得到的只是一个数据文件，如何判断采集到的数据是否正确，这里采用 Matlab 软件进行数据数据分析，用户可以通过修改我们提供的 matlab 源文件中的文件路径，进行数据分析。
	acx720_ad7606_ddr3_uart.rar：存放本次实验使用到的 vivado 工程。其中包括各个模块的源文件代码：串口接收数据模块、串口指令接收和指令解析模块、数据转换模块、ddr3 的含 fifo 的 2 端口封装模块、串口数据发送模块、AD7606 控制器模块等。
	data：测试数据文件夹，用来存放本次实验过程中，生成的数据文件，用户可以使用 Matlab 软件对我们提供的测试数据文件进行数据分析对比。

1.2. 实验平台说明

1. ACX720 开发板：Xilinx Artix-7 系列 FPGA 芯片，DDR3，串口。
2. 数据采集模块：ACM7606 模块。
3. 信号发生器：输出信号给数据采集模块 ACM7606。

1.3. ACM7606 数据采集模块简介

ACM7606 数据采集模块使用的是 ADI 公司的 16 位 8 通道同步采样模数转换器 AD7606，模块图如下图 1.1 所示。



图 1.1 ACM7606 模块图

AD7606 是 16 位 8 通道同步采样模数数据采集系统 (DAS)。内置模拟输入箝位保护、二阶抗混叠滤波器、跟踪保持放大器、16 位电荷再分配逐次逼近型模数转换器 (ADC)、灵活的数字滤波器、2.5V 基准电压源、基准电压缓冲以及高速串行和并行接口。AD7606 采用 5V 单电源供电，可以处理 $\pm 10V$ 和 $\pm 5V$ 真双极性输入信号。同时所有通道均能以高达 200kSPS 的吞吐速率采样。输入箝位保护电路可以耐受最高达 $\pm 16.5V$ 的电压。无论以何种采样频率工作，其模拟输入阻抗均为 $1M\Omega$ 。采用单电源工作方式，具有片内滤波和高输入阻抗，因此无需驱动运算放大器和外部双极性电源。AD7606 抗混叠滤波器的 3dB 截止频率为 22kHz；当采样频率为 200Ksps 时，它具有 40dB 的抗混叠抑制特性。

芯片对外提供 SPI 和并行的数字接口。当 AD7606 的 8 个通道全部以 200KPS 的最高速率进行转换时，数据输出速率达到 25.6Mbps，需要使用高性能 MCU 的 SPI 外设才能勉强该速率要求。因此可以使用 16 位并口来进行数据的传输，提高数据传输速率。当 AD7606 应用在 FPGA 系统中的时候，使用 SPI 串行接口和并行接口都能够轻松的满足数据传输的速率需求。当在 FPGA 系统上应用 AD7606 时，可以通过在 FPGA 上设计 AD7606 控制转换逻辑，将转换结果数据直接存储到片上的存储器如 FIFO 或者 RAM 中，也可以存储到 FPGA 片外的存储器如 SRAM 或 SDRAM 中，然后由其他主控芯片如 MCU 或 DSP 读出，或者直接在 FPGA 内部进行数据的运算和处理。当然，由于 FPGA 片上可以设计软核控制器，也可以直接使用软核控制器完成数据的处理和传输工作。

1.3.1. 功能框图

AD7606 的功能框图如下图 1.2 所示：

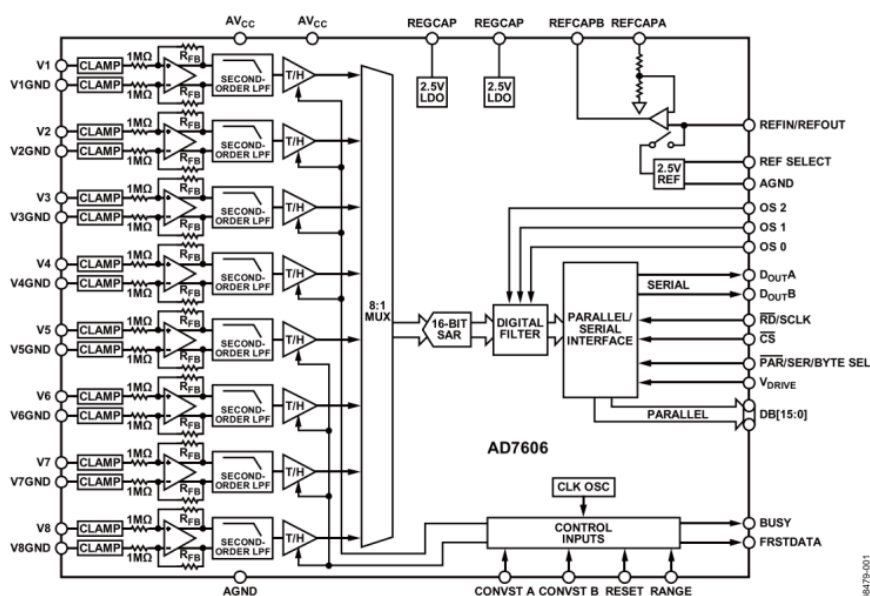


图 1.2 AD7606 功能框图

如上图所示，采集到的数据在经过稳压滤波和采样保持后通过 8 选 1 多路选择器被分别送入到 16 位逐次逼近型 ADC 芯片 AD7606 中进行转换，最后经由数字滤波后输出，当数据以串行模式输出时，数据会从 DoutA、DoutB 中输出，如果数据是以并行方式输出，那么数据将从 DB[15:0]中输出，下面为 AD7606 部分模块的说明介绍。

1.3.2. 模拟输入

AD7606 可处理真双极性、单端输入电压。RANGE 引脚的逻辑电平决定所有模拟输入通道的模拟输入范围。如果此引脚与逻辑高电平相连，则所有通道的模拟输入范围为 $\pm 10V$ 。如果此引脚与逻辑低电平相连，则所有通道的模拟输入范围为 $\pm 5V$ 。AD7606 的模拟输入阻抗为 $1M\Omega$ 。这是固定输入阻抗，不随 AD7606 采样频率而变化。AD7606 的输入结构如下图 1.3 所示，其各路模拟输入均含有箝位保护电路。虽然采用 5V 单电源供电，但此模拟输入箝位保护允许输入过压达到 $\pm 16.5V$ 。

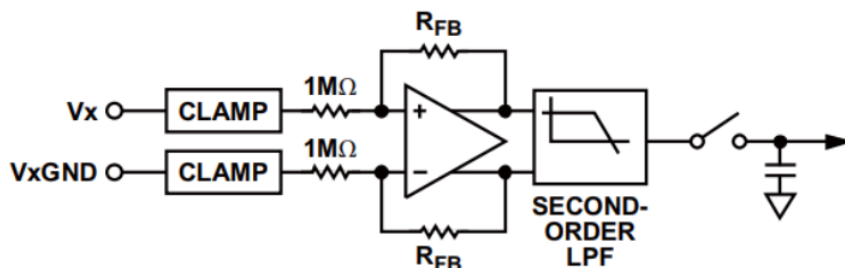


图 1.3 AD7606 模拟输入结构

1.3.3. 数字滤波器与过采样

店铺: <https://xiaomeige.taobao.com>

技术博客: <http://www.cnblogs.com/xiaomeige/>

官方网站: www.corecourse.cn

技术群组:

AD7606 内置一个可选的数字一阶 sinc 滤波器，在使用较低吞吐率或需要更高信噪比或更宽动态范围的应用中，应使用该滤波器。数字滤波器的过采样引脚 OS[2:0]控制（具体参考 AD7606 数据手册）。OS2 为 MSB 控制位。OS0 为 LSB 控制位，下表 1- 1 提供了用来选择不同过采样倍率的过采样位解码。

表 1- 1 不同过采样倍率的过采样位解码

OS[2:0]	过采样倍率	5V 范围 SNR (dB)	10V 范围 SNR (dB)	5V 范围 3dB 带宽 (kHz)	10V 范围 3dB 带宽 (kHz)	最大吞吐量 CONVST 频率 (kHz)
000	No OS	89	90	15	22	200
001	2	91.2	92	15	22	100
010	4	92.6	93.6	13.7	18.5	50
011	8	94.2	95	10.3	11.9	25
100	16	95.5	96	6	6	12.5
101	32	96.4	96.7	3	3	6.25
110	64	96.9	97	1.5	1.5	3.125
111	无效					

OS 引脚在 BUSY 下降沿锁存，从而设置下一个转换的过采样倍率，如下图 1. 4 所示，如果 OS 引脚选择过采样倍率 8，则下一个 CONVST x 上升沿采集各通道的第一个采样点，一个内部产生的采样信号采集所有通道的其余 7 个样点，然后对这些样点求平均值，以改进 SNR 性能。开启过采样时，CONVST A 和 CONVST B 引脚必须连在一起驱动，转换过程中 BUSY 保持高电平的时间会延长。BUSY 保持高电平的实际时间取决于所选的过采样倍率，过采样倍率越高，则 BUSY 保持高电平的时间或总转换时间越长。

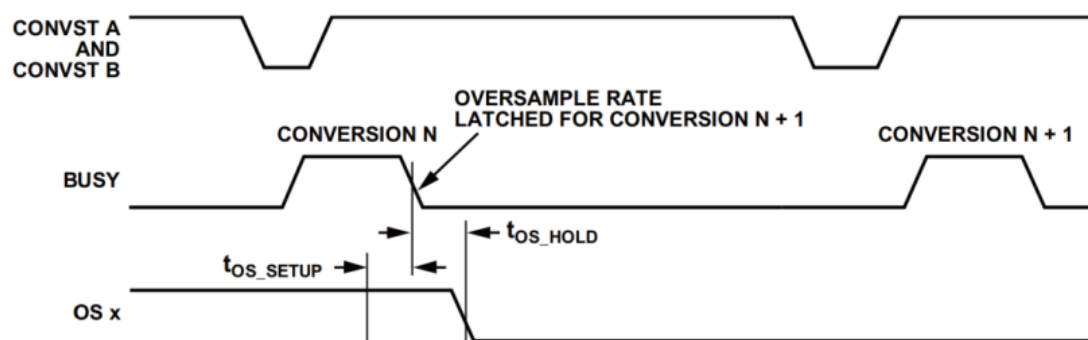


图 1. 4 OS 引脚在 BUSY 下降沿锁存时序示意图

1.3.4. 工作时序

AD7606 根据采样方式不同具有多种驱动时序，本次采用的为并行输出（即 8 个 16 位的数据通过 16 根并行线一个接着一个输出），转换后读取模式。其时序图由两部分组成：完成 AD 转换和读取 AD 数据。其中的时间可以参考 ADI 公司的手册。当 CONVST A 和 CONVST B 通道都变为上升沿时，BUSY 信号转变

为高电平，代表转换开始，知道 BUSY 的下降沿到来，代表数据已经转换完成，正在锁存至输出数据寄存器中，当 \overline{CS} 变为下降沿时，数据将会被输送到总线上。并行工作模式下，当 \overline{CS} 和 \overline{RD} 都为低电平时，会使能总线，将转换结果输出到并行数据总线上，当 V1 转换结果开始输出之后，FRSTDATA 会随后转变为高电平，表示输出数据总线可以提供 V1 的结果。并行模式下，每次数据的输出为 16 位，对应一个通道。

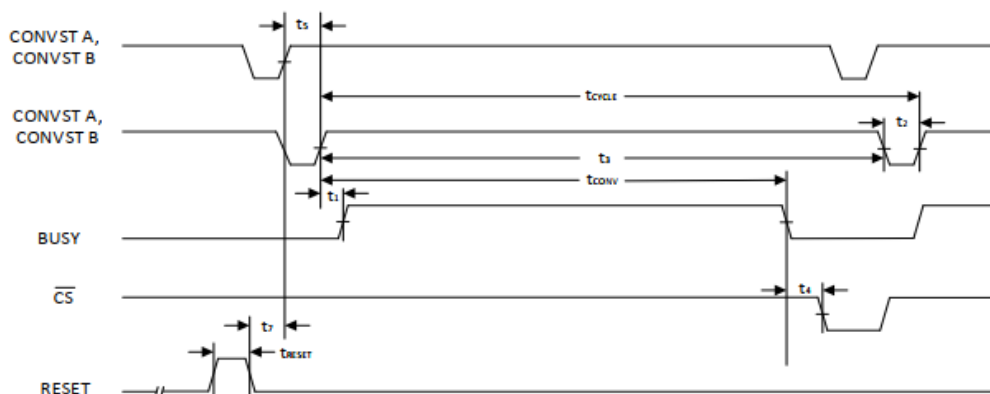


图 1.5 CONVST 时序—转换之后读取

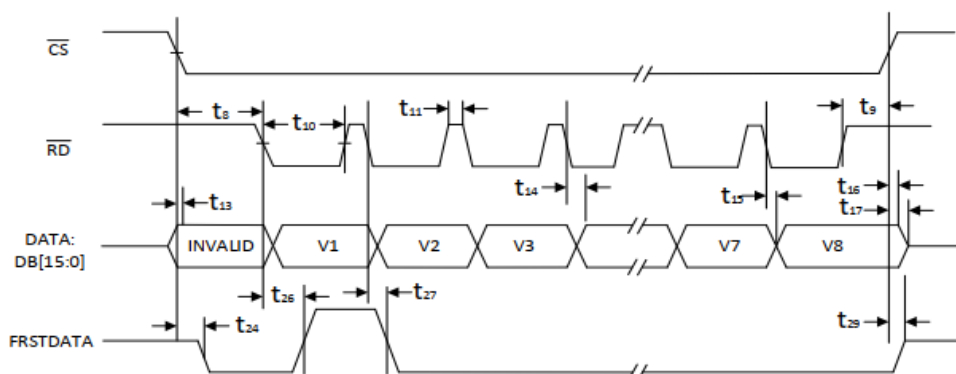


图 1.6 并行模式，独立的 \overline{CS} 和 \overline{RD} 脉冲

1.4. 实验内容概述

1.4.1. 实验总体设计

本案例基于 Xilinx Artix-7 系列 FPGA，结合 ADI 公司的 16 位 8 通道并行采样 ADC 芯片，实现了对 AD7606 型 8 通道 16 位 ADC 的数据转换控制并输出，数据采集完成后存储在 DDR 中，并通过串口传输，用户可以在电脑上通过串口调试工具对采样到的数据进行查看并通过指令对 ADC 芯片进行配置，控制其采样频率，数据采样个数以及通道数，并将输出的数据导出，进行进一步的数据处理分析。实验总体设计如下图 1.7 所示：

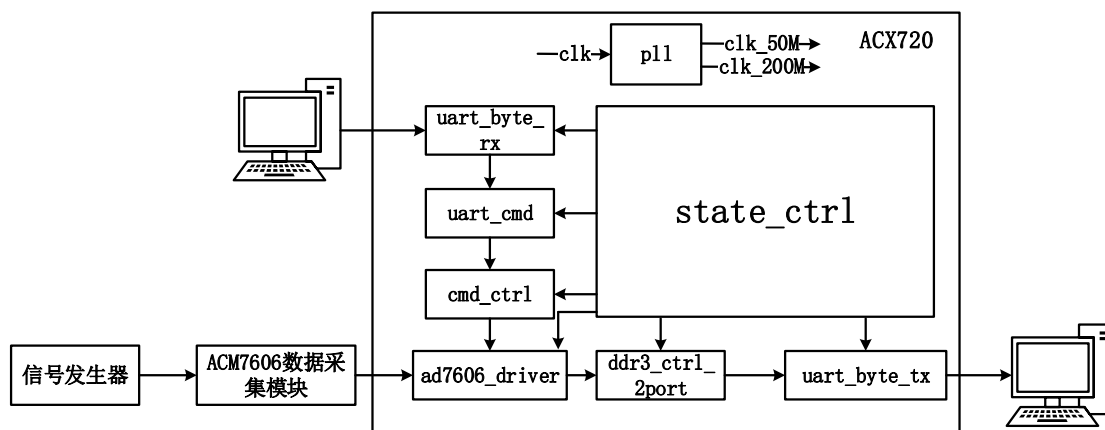


图 1.7 实验总体设计

上述图中各个模块的功能说明如下，具体模块代码请自行查看工程中对应的源文件：

1. `uart_byte_rx`: 串口接收数据模块，接收串口调试助手传输过来的指令。
2. `uart_cmd`: 从接收的串口数据中识别出指令类型。
3. `cmd_ctrl`: 接收到的指令进行翻译拆解，指令分类。
4. `ad7606_driver`: `acm7606` 控制器模块，该控制器实现了对 `AD7606` 型 8 通道 16 位 ADC 的数据转换控制并输出。使用该控制器时，用户无需关心 `AD7606` 的具体控制时序，一切都在控制器内部完成，用户只需要像使用并行 ADC 一样取用数据即可。
5. `ddr3_ctrl_2port`: `ddr3` 的含 `fifo` 的 2 端口封装模块，主要负责整个数据的存储功能。
6. `uart_byte_tx`: 串口发送模块，将最终采集到的数据通过串口发送出去。
7. `state_ctrl`: 状态机模块，协调各个模块的信号控制，程序状态的总控制模块。
8. `p11`: 锁相环模块，50M 时钟输入，输出 200M 的时钟给 `DDR3`，50M 的时钟给其他模块。由于我们在设计时，对于时钟频率没有要求的模块，都可以采用 50MHz 的频率进行设计，而 `DDR3` 驱动模块的工作时钟频率最低也得 200MHz 才能支持正常工作，所以我们需要使用锁相环模块完成整个工程的频率配置，以实现在工程中引入两种时钟频率的设计需求。

1.4.2. `ad7606_driver` 模块说明

该控制器接口总共分为 4 个类，第一类为时序逻辑模工作所必须的时钟和复位信号 (`clk`、`Reset_n`)，第二类是 `AD7606` 控制器与 `AD7606` 芯片管脚相连的各

种功能控制和数据信号，第三类是设置控制器工作状态/工作数据的用户控制接口，第四类是控制器的结果输出接口。对于用户来说，只需要关注第三类和第四类接口的使用，即可快速高效的使用该控制器来控制 AD7606 芯片完成数据转换。ad7606_driver 接口图如下图 1.8 所示。

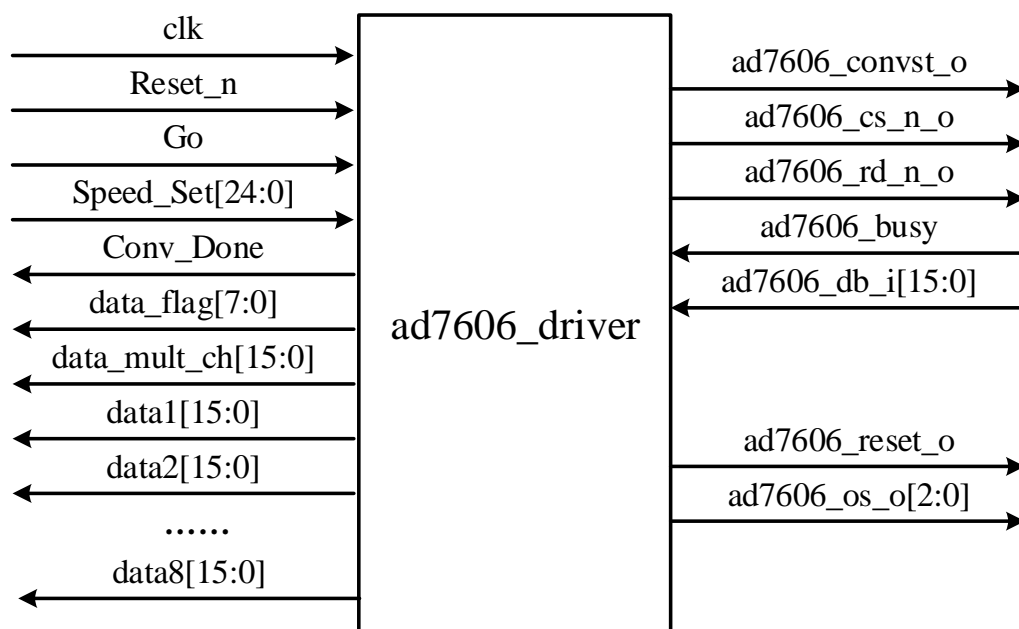


图 1.8 ad7606_driver 接口图

端口功能说明如下表 1-2 所示。

表 1-2 ad7606_driver 端口说明表

类	信号名	位	功能简介
控制信号	Clk	1	系统时钟，为了让采样速率准确，要求为 50MHz。
	Reset_n	1	系统复位，低电平复位。
	Go	1	采样使能信号，为高电平就使能采样，低电平则在已经开始的一轮采样结束后，停止下一次采样。
	Speed_Set	25	采样速率控制端口， $Speed_Set = 1000000000/20/speed - 1$ 。
数据结果输出端口和标志	Conv_Done	1	一次采样完成标志信号，单时钟周期脉冲信号。每次 8 个通道结果都输出后，产生一个高脉冲信号。
	data_flag	8	转换结果有效标志信号，因为 AD7606 有 8 个通道，转换结果是依次输出，并非同时的，所以设置 8 个 Flag 信号，每个通道的结果就绪之后，就产生一个 Flag 信号，通知外部可以取用。另外，如果只关心其中的部分通道，则只需要关心 data_flag 中对应的位即可。
	data_mult_ch	16	多通道数据输出端口，该通道 16 位，在不同的时刻，输出不同通道的转换结果，使用时，与 data_flag 信号配合，data_flag 的哪一位出现高脉冲，则代表当前 data_mult_ch 的值为该通道的转换结果。该端口设计的目的是用于往 FIFO、RAM 等存储器中存储结果时使用。
	data1..8	16	8 个通道的采样结果输出端口，每个端口分别对应一个模拟通道的采样结果，使用时与 data_flag 信号配合，每当 data_flag 中的某一位为 1 时，则对应的通道上的 16 位采样结果已经就绪，可以使用。这些端口主要用

信号			于每个通道的数据需要分别应用的场合。
ADC 芯片控制信号	ad7606_cs_n_o	1	ad7606 芯片选中控制信号，可以从 AD7606 中读取转换结果时，需要使该信号为低电平。
	ad7606_rd_n_o	1	ad7606 转换结果读取信号，该信号的下降沿，AD7606 将特定通道的采样结果送到 16 位数据线上，供外部读取。外部可以在 rd_n 信号的上升沿读取 16 位数据线上的结果。
	ad7606_busy_i	1	ad7606 转换状态标志信号，为高电平则表明 ad7606 当前仍处于转换状态，结果没有更新，如果此时读取，读取的结果就还是前一次的采样转换结果。需要待该信号变为低电平之后，再读取 ad7606 中的数据。
	ad7606_db_i	16	ad7606 的 16 位数据线，读取时，输出对应通道的转换结果。
	ad7606_os_o	3	ad7606 过采样控制信号，使用过采样可以进一步提高 ad7606 的采样精度，使用过采样会降低 ad7606 的有效转换速率，关于过采样的功能和使用方法，可以参考 ad7606 的 datasheet，默认为 0，则表示不使用过采样。能够运行在最高的转换速率（200Ksps）。
	ad7606_reset_o	1	ad7606 的复位信号，复位 ad7606 内部各个功能单元的工作状态。
	ad7606_convst_o	1	ad7606 转换开始信号，该信号的上升沿启动 ad7606 内部的采样转换逻辑开始新一轮的采样转换。

1.5. 实验功能测试

1.5.1. 硬件连接

要验证本次实验，首先第一步就是对本次实验的整个系统硬件进行连接，本次工程硬件具体的连接如下图 1.9 所示。

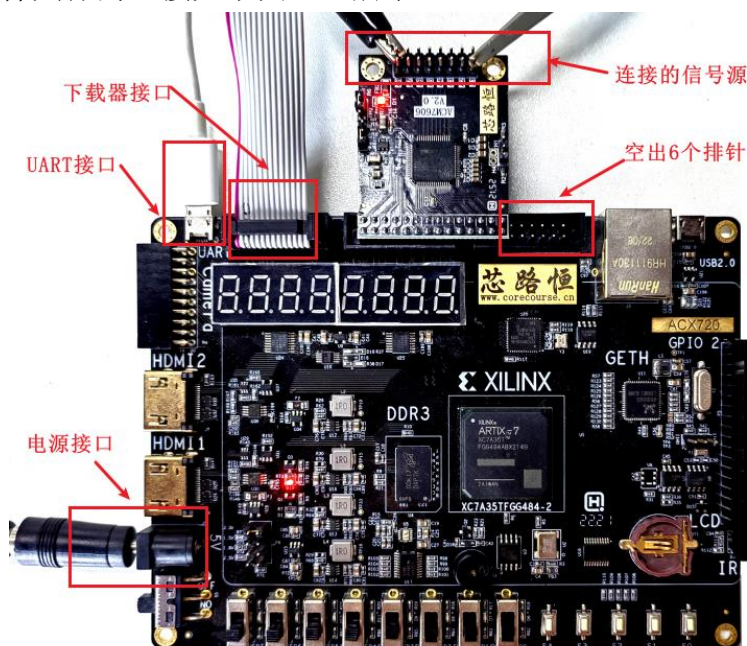


图 1.9 整体硬件连接图

本次使用的信号源是通过信号发生器产生的，一端接 A1，一端接地。注意串口必须连接在 ACX720 板子的左侧接口，AD7606 连接正确后右边会多出 6 组排针，由于视觉缘故，这里很容易连接错误，连接错误后串口猎入上很容易出现有发送而无接收的情况，电源线与下载器连接如上图所示，在硬件连接好后打开开关就可以下载程序。

1.5.2. 下载 bit 文件

打开本次实验的 vivado 工程，下载本次实验的 bit 文件，用户可以直接将我们提供的 bit 文件夹下的文件烧写进板中，也可以打开工程一步步下载，操作如下图所示 1.10 所示。

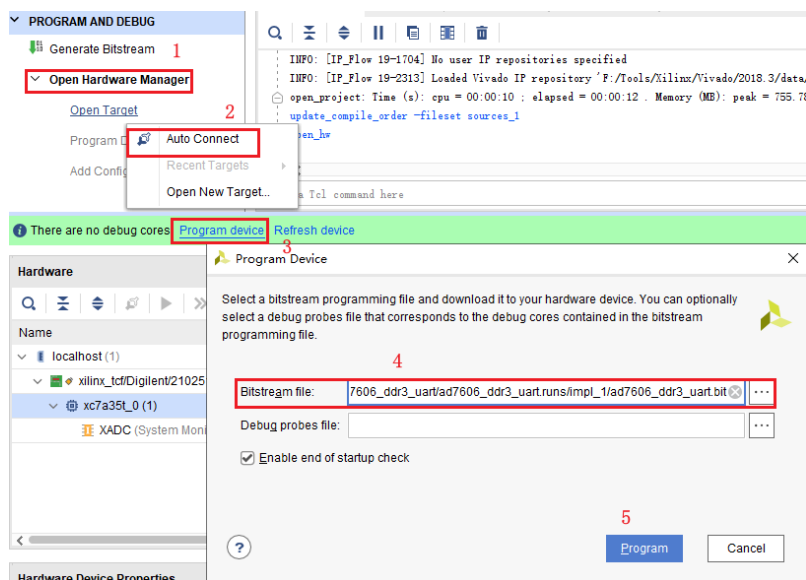


图 1.10 下载本次实验的 bit 文件

程序烧写完成后，开发板的 led0 和 led1 的灯会点亮，如下图所示 1.11 所示，此时表明：锁相环工作正常，ddr3 初始化正常。

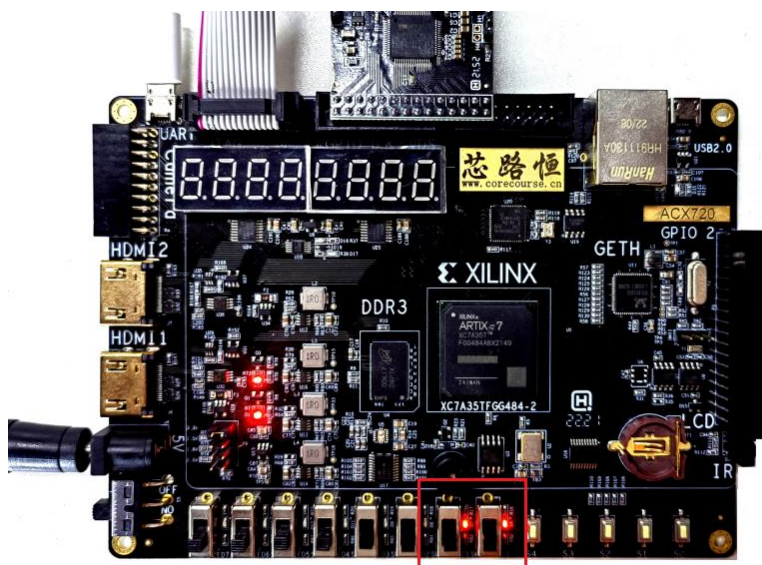


图 1.11 程序下载完成之后，开发板现象

1.5.3. 配置 ADC

本次实验是通过发送指令的方式来配置 ADC，下面将介绍如何通过串口助手发送指令以及指令代表的含义。

1.5.3.1. 发码指令设置

我们发码指令参数，总共分为以下类别：

1. 设置采样通道 (Channel 寄存器, 地址为 1): 如 55 A5 01 00 00 00 01 F0, 对通道 1 进行采样。通道设置寄存器, 共 8 位, 对应 8 个通道的数据存储开关, 如果某通道对应的设置位为 1, 则该通道的采样结果就会被存入 DDR 并通过串口发送。
2. 设置采样个数 (DataNum 寄存器, 地址为 2): 如 0x55 0xA5 0x02 0x00 0x00 0x3E 0x80 0xF0, 设置的采样个数为 16000 个 (0x3E80)。注意, 该寄存器设置的总共采集的数据个数, 假设设置采集 100 个数据, 而且设置 ChannelSel 为 0000_0011, 则实际每个通道采样的次数就是 50, 2 个通道的数据加起来就是 100 个。假设设置采集 100 个数据, 而且设置了 ChannelSel 为 0011_0011, 则实际每个通道采样次数就是 25, 4 个通道加起来采集 100 个数据。
3. 设置采样速率 (ADC_Speed_Set 寄存器, 地址为 3): 如 0x55 0xA5 0x03 0x00 0x00 0x01 0xF3 0xF0, 设置采样速率为 100K。该寄存器用来设置 ADC 每多久执行一次转换, 由于 ADC 的最大采样速率为 200Ksps, 所以可以通过设置该寄存器的值来让 ADC 的采样速率在 1~200Ksps 范围内调整, 以适应不同的应用场景。该寄存器的值与采样速率关系为:

ADC_Speed_Set=1000000000/20/speed-1，其中 speed 就是实际要设置的采样速率。

- 启动传输（RestartReq 寄存器，地址为 0），重新开始采集请求寄存器，向该寄存器写入任意值即可启动新一轮的采样存储传输。

1.5.3.2. 设置串口助手

打开串口猎人，注意看有没有检测到串口，如果没有就看看是不是串口连接线连接有问题或者是否没装驱动，接着波特率一栏设置其为 115200，如下图 1.12 所示。

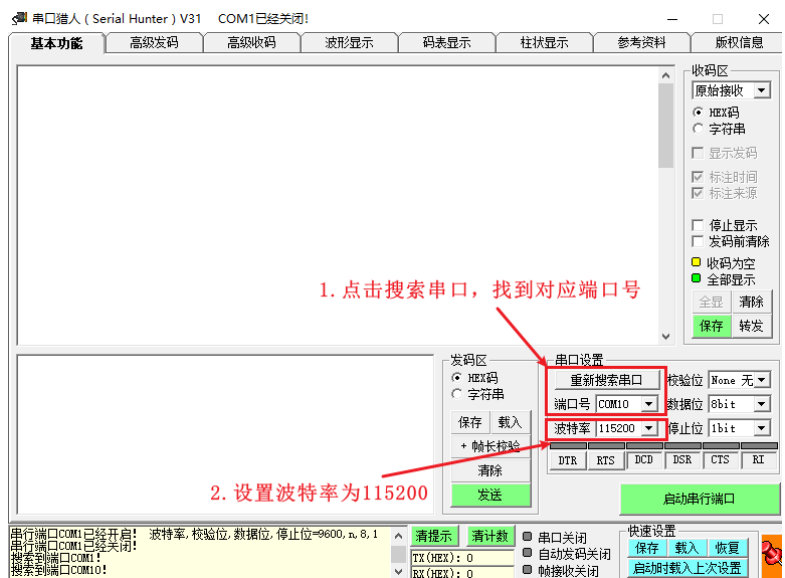


图 1.12 选择端口号和波特率

接着打开高级发码界面，发送对应指令，比如，举例本次实验发送如下表 1-3 所示的指令。

表 1-3 发送指令说明表

指令定义	指令意义
55 A5 02 00 00 40 00 F0	采集 16384 个数据（串口将会发送 32768 个数据）
55 A5 01 00 00 00 01 F0	选择通道 1
55 A5 03 00 00 00 F9 F0	设置采样速率为 200K
55 A5 00 00 00 00 00 F0	启动传输

串口猎人高级发码设置界面如下图 1.13 所示。注意，这里的循环次数设置为 1，不然会一直循环发送。

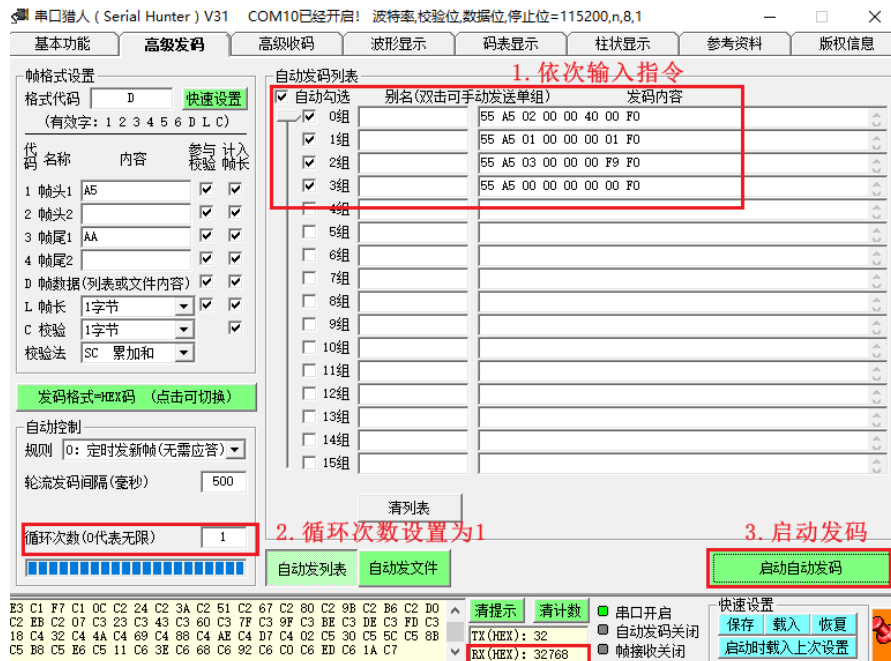


图 1.13 设置串口猎人发送指令

发送完指令后 FPGA 传回的数据，注意这里由于数据过多而被隐藏，我们需要点击全显，让其全部显示出来，由于数据过多，为了能够直观体现数据的完整性，我们将数据以.txt 格式保存，通过 MATLAB 将数据转换为图像，操作如下图 1.14 所示。

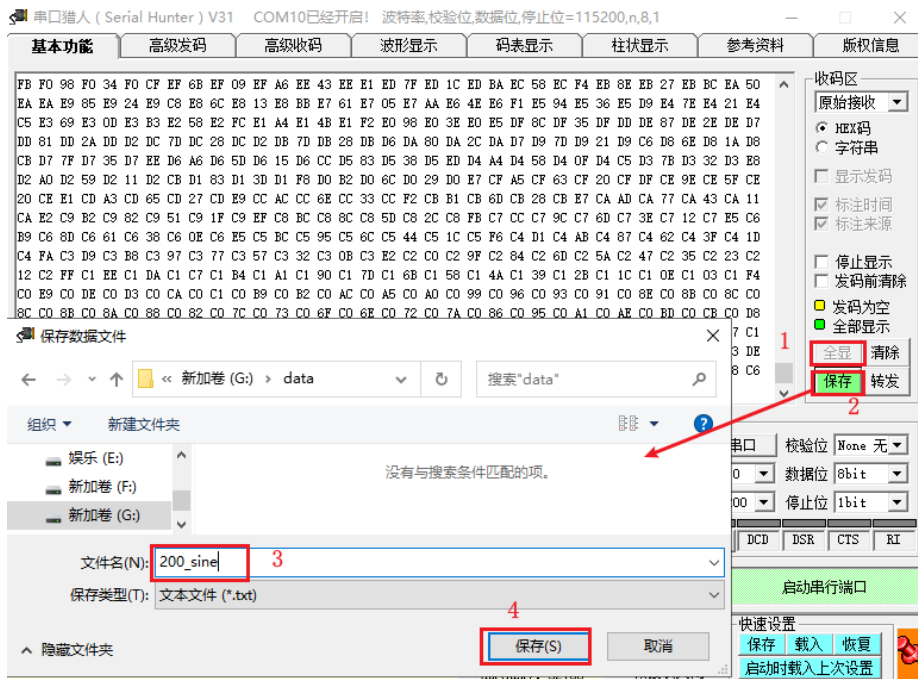


图 1.14 保存接收到的数据

1.5.4. MATLAB 图像绘制

店铺: <https://xiaomeige.taobao.com>
 技术博客: <http://www.cnblogs.com/xiaomeige/>

官方网站: www.corecourse.cn
 技术群组:

双击我们提供的 ADCdata_to_wave_v1_1.m 文件，在打开方式里选择以 MATLAB 打开，然后修改文件路径，随后运行便可以直观的看到数据是否正确，操作如下图 1.15 所示。

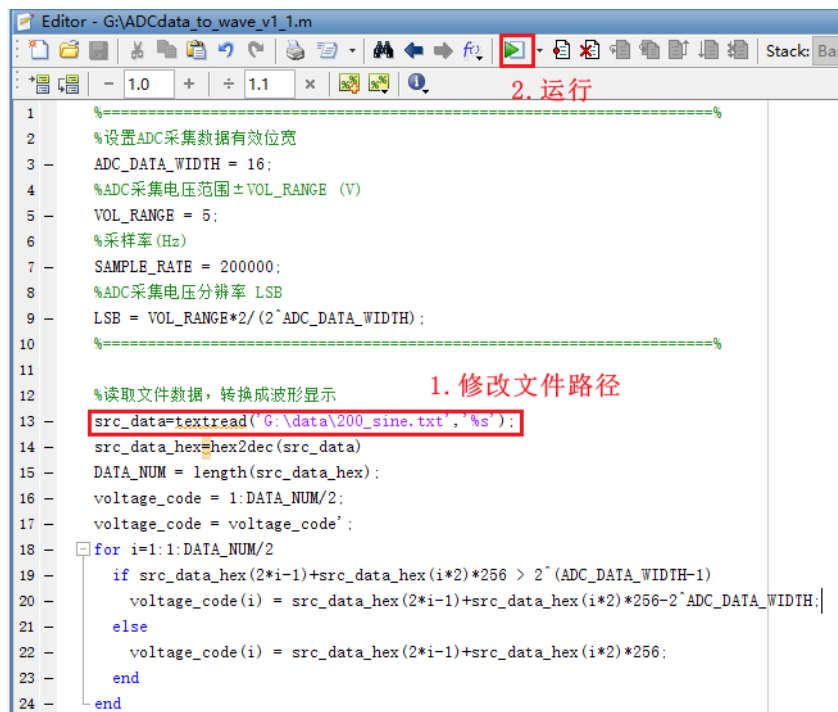


图 1.15 Matlab 设置界面示意图

运行之后，得到的波形如下图 1.16 所示，信号源输出的波形如下图 1.17 所示。对比可以看出波形数据基本一致。

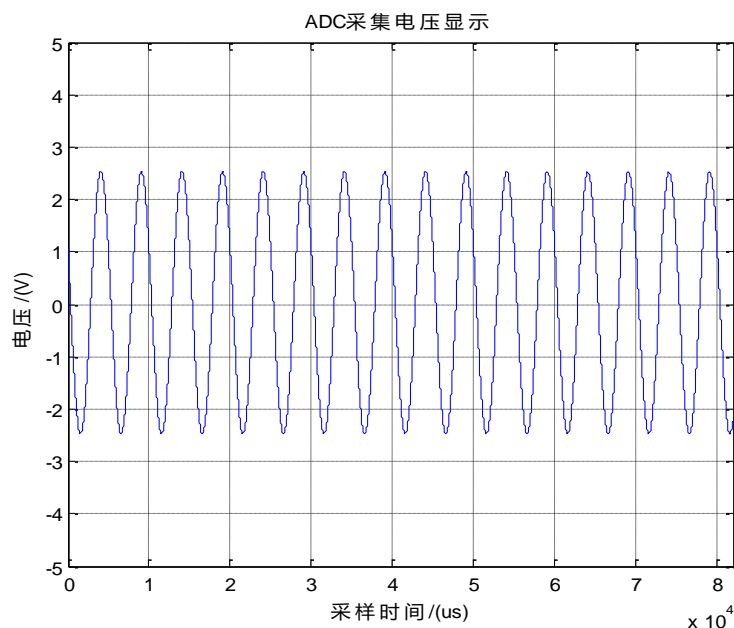


图 1.16 MATALB 分析采集到的数据得到的波形

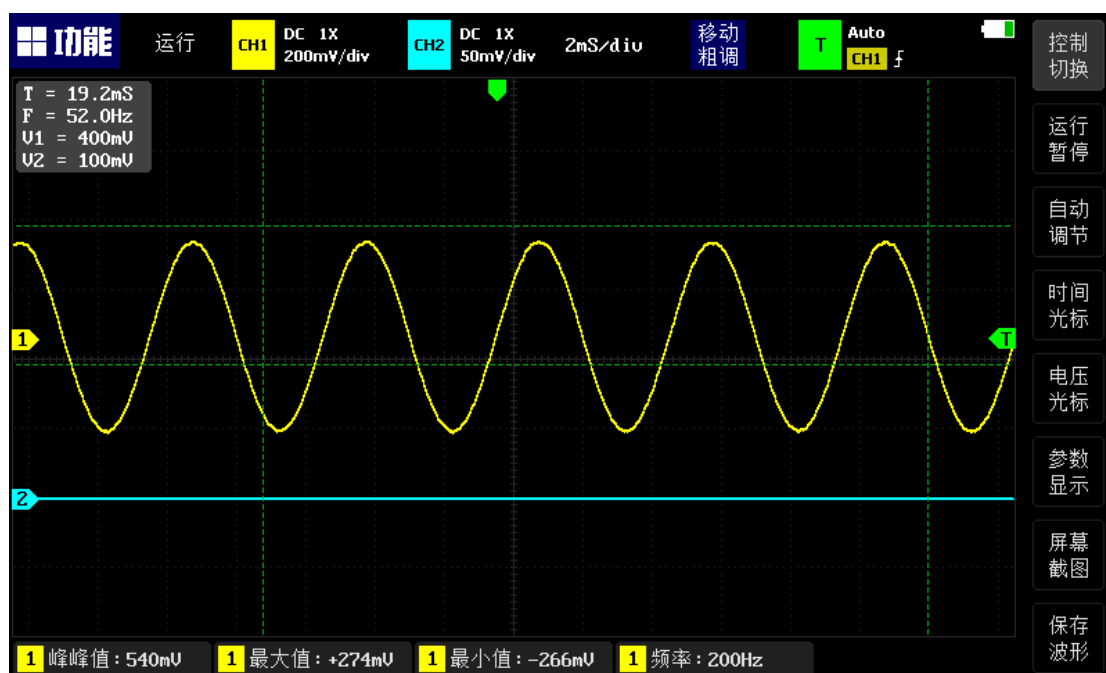


图 1.17 信号源输出波形图

1.6. 简易数据分析

为了证明我们的工程在不同设置状态下工作的可靠性，我们给出一些实验数据结论的波形进行对照参考，对应就是 data 文件夹中的实验数据文件。

- 200hz，方波和三角波波形图如下图 1.18 和图 1.19 所示。

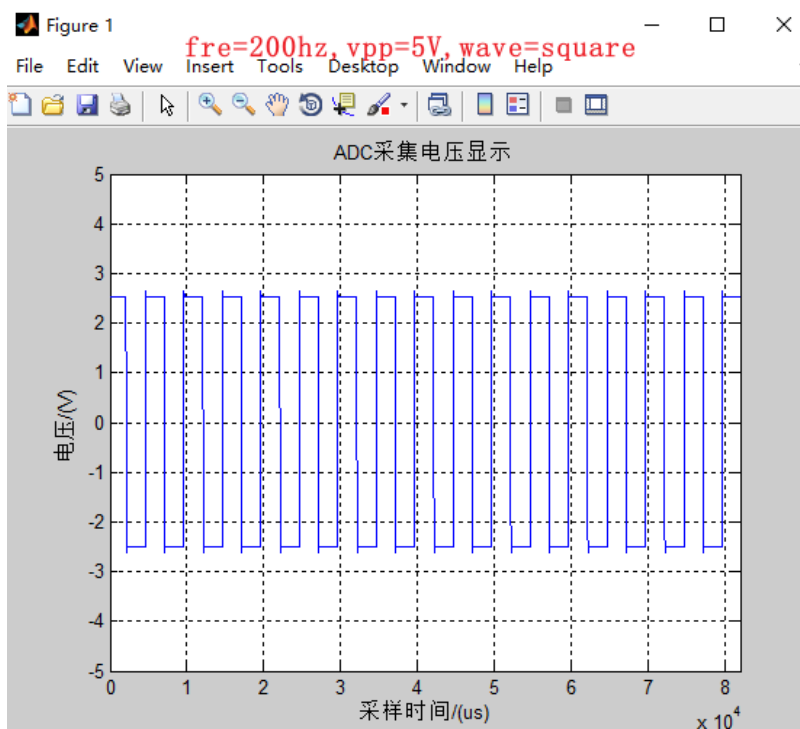


图 1.18 200hz 方波波形图

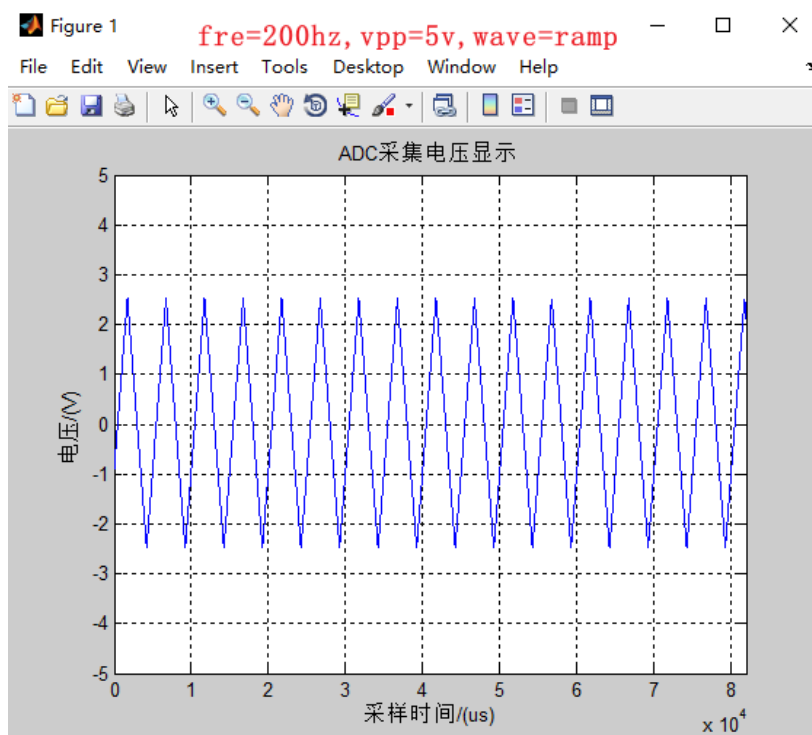


图 1. 19 200hz 三角波波形图

2. 不同频率正弦波波形图，频率分别为 100hz、500hz、1khz。波形图如下
图 1. 20、图 1. 21、图 1. 22 所示。

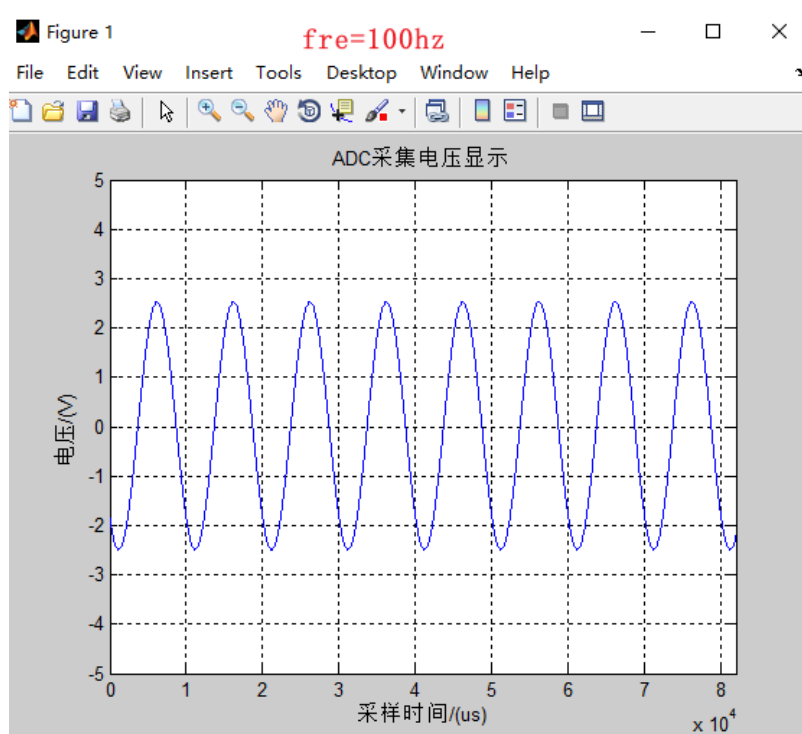


图 1. 20 100hz 正弦波波形图

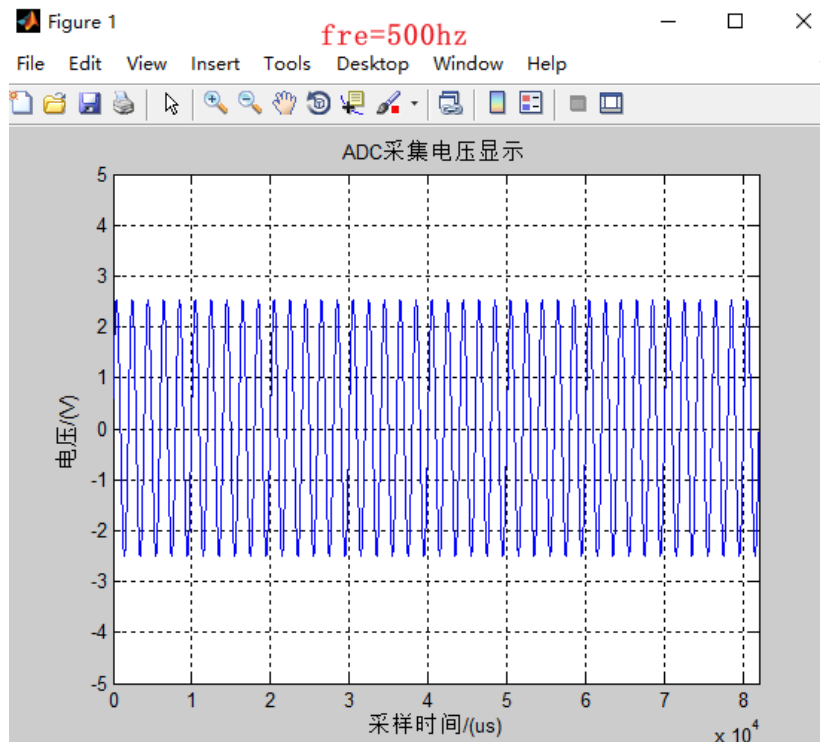


图 1. 21 500hz 正弦波波形图

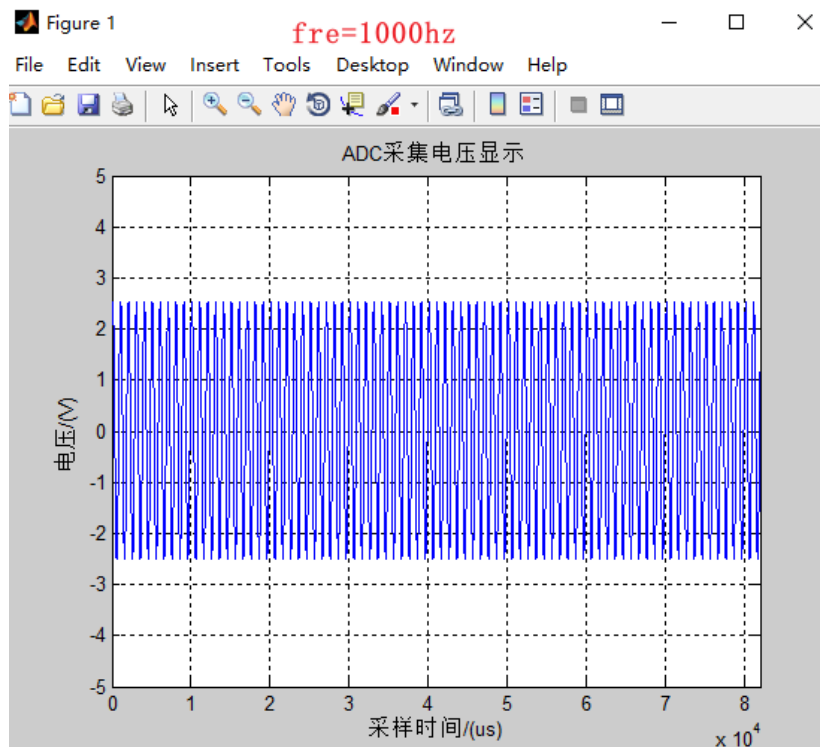


图 1. 22 1khz 正弦波波形图

本次实验仅对不同波形以及不同频率的波形进行了数据分析,其他功能读者可以自行修改测试。