

双 DSP 嵌入式系统远程 SCI 通信烧写技术

王璐

(中国科学院长春光学精密机械与物理研究所 吉林省长春市 130000)

摘要: 本文在双 DSP 的嵌入式硬件基础上, 设计了一种基于 SCI 通信的远程烧写 HEX 文件进行程序升级的方案。该方案通过上位机选择后, 待升级的 DSP 基于 SCI 通讯读取待升级的 HEX 文件, 调用 TI 的 API 库函数对存储空间进行擦除、烧写和校验等操作, 完成程序的更新。经实验证明, 该方法操作简单、有效可行。

关键词: 远程烧写; SCI 串口; 双 DSP; API 函数

1 引言

数字控制处理器 (DSP) 具有高速的数据运算能力, 功能丰富, 接口多样, 具有强大的控制能力, 而且易于开发。基于 TMS320F28335 DSP 的嵌入式系统, 更是广泛应用于电机控制、通信、医疗、工业控制等领域^[1]。在伺服控制上, 为了提高控制精度往往需要大量复杂的算法和更高的采样频率, 这就对 DSP 的运算能力提出了更高的要求, 双 DSP 并行处理的嵌入式系统是一种解决方案^{[2][3]}。传统的程序升级方式依赖于 JTAG 接口, 通过仿真器连接计算机访问 DSP 的存储空间, 进行仿真和烧写。这种连接方式稳定, 方便操作, 但价格昂贵, 使用场所受限, 适用于实验室设计开发阶段^[4]。嵌入式系统设备交付后实际使用环境往往复杂多变。可能面临无法插拔仿真器, 或者出现 JTAG 接口距离过长无法烧写成功的情况。限制了 DSP 的进一步应用, 对系统的可维护和易更新等方

面提出了进一步的要求。所以开发出稳定, 并且能够远距离传输, 不受 JTAG 仿真器接口限制的程序升级方法非常重要。

文献^{[5][6]}基于 TMS320F2812 内部 FLASH 提出一种在线烧写技术, 提高了系统的可维护性。文献^{[7][8]}通过配置 GPIO84~GPIO87 引脚的电平选择启动方式, 选择从 SCI、SPI 或 CAN 总线等方式启动。烧写完毕后将 GPIO87 重新拉高, 重新启动 DSP 即可实现程序的升级运行。文献^[9]通过 SCI 串口分别与上位机和辅助芯片 STM32 接收指令和用户程序实现烧写。文献^[10]通过将待升级程序复制到 FLASH 备份扇区, 保证程序完成性, 能够在烧写意外断点情况下修复并运行。文献^[11]设计一种基于 GPRS 通信模块的系统结构实现了远程无线通讯。文献^{[12][13]}基于 CAN 总线通讯设计了一种远程程序方案。

本文基于一种双 DSP 伺服控制硬件电路上进行远程程序升级

开发环境是由美国国家仪器 (NI) 公司研制开发, Labview 与其他计算机语言的显著区别是: 其他计算机语言都是采用基于文本的语言产生代码, 而 Labview 使用的是图形化编辑语言 G 编写程序, 产生的程序是框图的形式^[9]。G 代码相对于其他语言较容易学习, Labview 开发环境分为前面板和程序框图。前面板: 根据用户的要求和喜好选择相应合适样式的控件并进行摆放, 设置背景颜色, 字体颜色及大小等, 在此界面实现人机交互对话功能。程序框图: 根据设计目的对前面板确定的控件选择相应的函数结构并连线编程。一般情况下先在前面板选择控件, 后在程序框图根据功能进行编程。Labview 开发环境自带即时帮助和范例, 可随时查看各函数含义帮助理解使用, 同时也方便系统学习编写程序; 前面板可自行美化界面提高观赏性。相对而言 Labview 开发环境功能较为强大, 在本次设计中只使用到了进行数据传输的串口通信、TCP/IP 远程通信、数据分析、显示、存储、程序调试等小部分功能, 还有更多方面值得研究学习。

用户界面就是 VI 程序的前面板, 使用者可直接在面板进行操作控制, 方便简单。为实现远程通信, 将上位机部分分为服务器和客户端两部分, 服务器置于机房; 客户端远程监控, 处在同一局域网即可。客户端用户界面如图 4 所示。

4 结论

本文采用 Labview 设计操作界面, 采用 STC12C5A60S2 单片机作为主控器, 传感器 DHT12 采集温度、湿度, LT163 单相互感式计量模块采集电压、电流, 结果表明该系统能够对机房设备的运行状态、温湿度、供电电压电流等参数进行实时的监控, 实现对机房的远程智能化管理。

参考文献

- [1] 周小凤, 郭晓勇. 基于 Labview 和 GPRS 的供水管网远程监控系统设计 [J]. 包头职业技术学院学报, 2018, 19 (03): 6-8.
- [2] 彭举. 智能无人值守设备机房监控系统 [J]. 广东公路交通, 2018, 44 (04): 196-198.
- [3] 侯山山, 黄丹平, 张芯豪, 徐同旭, 郭康, 李志荣. 基于 Labview 智能曲房测控系统研发 [J]. 食品工业, 2018, 39 (05): 234-238.
- [4] 祝奔奔, 万舟. 基于 Labview 的光伏电站远程监控系统设计 [J]. 软件, 2017, 38 (11): 49-54.
- [5] 王珍凤, 杨路. 基于 Labview 与无线传感器网络的实验室远程监控系统设计 [J]. 洛阳理工学院学报 (自然科学版), 2017, 27 (03): 65-68+93.
- [6] 徐伟, 俞春飞, 艾伟清. 基于 Labview 的远程温湿度监控系统设计 [J]. 常熟理工学院学报, 2017, 31 (04): 71-74.
- [7] 王静, 周静. 无人值守电力机房环境监控系统设计 [J]. 现代商贸工业, 2017 (05): 188-190.
- [8] 薛航. 基于 Labview 的远程环境监控系统研究 [J]. 科技创新与应用, 2016 (31): 103.
- [9] 李光明, 孙英爽, 党晓娟. 基于 Labview 和 Arduino 的远程监控系统设计与实现 [J]. 计算机测量与控制, 2015, 23 (10): 3522-3524+3528.

作者简介

蒋久芳, 大学本科学历, 工程师。研究方向为电气自动化。

的方案设计。采用一种基于 SCI 通讯的程序升级方案,无须仿真器,同时将控制函数嵌入到用户程序中,进行控制函数的搬移,实现程序的升级。

2 API 库函数

可通过 TI 自带的 Flash API (The Flash Application Program Interface) 功能库可实现脱离仿真器 JTAG 的多种程序升级方式,无论采用什么方式烧写都需要用 Flash API 库内的算法实现对 Flash 区域的操作和编辑^{[14][15]}。API 库主要由以下几个函数组成:

API 库的函数 APIVersion 和 APIVersionHex 用于确定当前 API 库文件的版本,前者返回浮点型的数,后者返回 Hex 的数,用于判断当前 API 库的版本防止由于版本错误导致的升级失败。

擦除函数仅对 Flash 区域进行操作,且当烧写进行时无法进行擦除操作,擦除操作将 Flash 区域的数据全部写 0xFFFF,擦除 Flash 时操作单元时一个扇区,不可对扇区内的某一位再进行操作,没有擦除指令的扇区内容保持不变。Flash28335_Erase(Uint16SectorMask,FLASH_ST*FEraseStat); 函数中的 SectorMask 定义标示位来指示擦除的扇区;定义 FLASH 结构的状态指针 FeraseStat,用于确定执行擦除操作后状态返回值,判断操作是否成功。

擦除程序后的各个扇区数据均为 0xFFFF,烧写程序的实质就是对扇区内的 1 进行写 0 操作,烧写程序函数为:

```
Flash28335_Program(Uint16*FlashAddr,Uint16*BufAddr, Uint32
Length, FLASH_ST *FProgStatus);
```

其中 *FlashAddr 指向烧写的数据存入 Flash 区域的第一个地址,*BufAddr 为缓冲区域指针,Length 为数据长度,*FprogStatus 表示为烧写的状态指针。

验证程序为烧写后的附加程序。API 库中有如下定义。

```
Flash28335_Verify(Uint16 *FlashAddr,
Uint16 *BufAddr, Uint32 Length,
FLASH_ST *FVerifyStat);
```

其中 *FlashAddr 指向烧写的数据存 Flash 区域的第一个地址,*BufAddr 为缓冲区域指针,Length 为数据长度,*FVerifyStat 表示为验证地址。

由于 API 库函数需要对 FLASH 区域进行操作,所以不可运行于在 Flash 区域,运行时需要将储存在 FLASH 区域的程序搬移到 RAM 中运行。几种搬移方式流程如图 1 所示。

API 程序的搬移方式有三种,一是通过 JTAG 引脚,依然需要仿真器。二是通过 SCI、SPI 等通讯的 A 通道实现烧写的控制程序,这种升级方法需要对相应的引脚进行拉高和拉低,并且需要严格用通讯的 A 通道进行升级,操作相对较为复杂,也不灵活。三是 API 函数代码嵌入到 FLASH 代码中,每次程序运行都将程序运行至 RAM 中去。

API 库函数搬移到 RAM 区运行可通过在 .cmd 工程文件的配置实现,程序文件中必须包含以下两个工程的语句 -lFlash28335_API_V210.lib(.econst) 库文件源代码 -l Flash28335_API_V210.lib(.text),是 API 库文件的参数。

```
Flash28_API:
{-l Flash28335_API_V210.lib(.econst)
-l Flash28335_API_V210.lib(.text)}
LOAD = FLASHA,
RUN = RAML0,
```

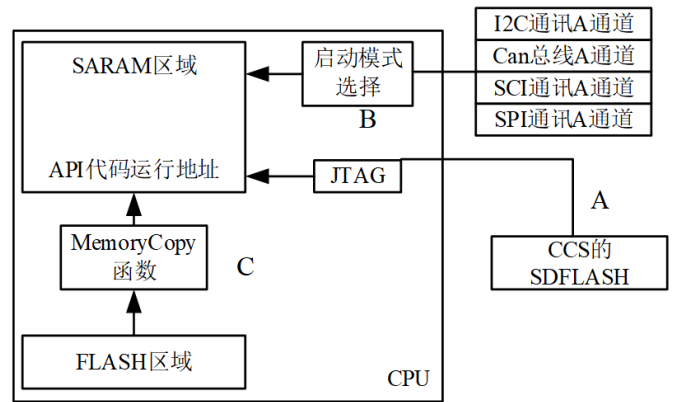


图 1: API 复制到 RAM 区的方式

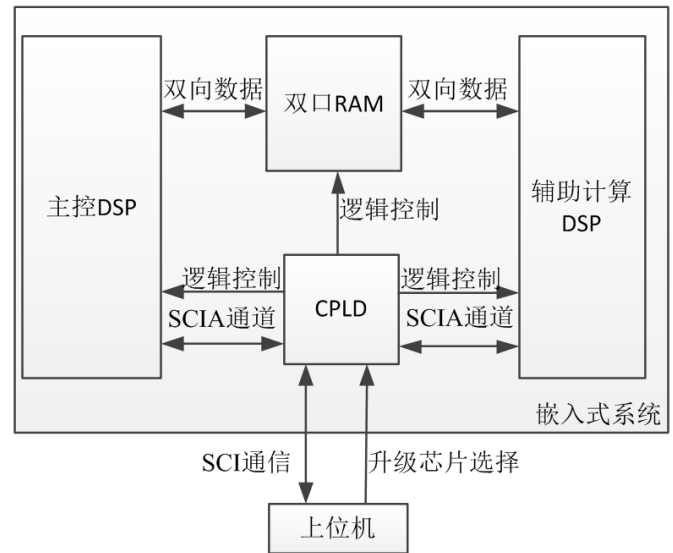


图 2: 双 DSP 系统嵌入式系统框图

```
LOAD_START(_Flash28_API_LoadStart),
LOAD_END(_Flash28_API_LoadEnd),
RUN_START(_Flash28_API_RunStart),
PAGE = 0
```

此段程序定义了 API 函数段装载在 PAGE0 的 FLASHA 中,运行地址在 PAGE0 的 RAML0 中。令编译器创建了变量分别指向该段的首地址、末地址和起始地址。上述 .cmd 文件中定义了起始终止和运行地址,这些均已在 API_Library.h 中进行了定义。

```
extern Uint16 Flash28_API_LoadStart;
extern Uint16 Flash28_API_LoadEnd;
extern Uint16 Flash28_API_RunStart;
```

在用户程序 .c 文件中,添加代码进行如下调用和定义。即可实现 API 程序的搬移到 RAM 中。

```
#include Flash2833x_API_Library.h
MemCopy(&Flash28_API_LoadStart,&Flash28_API_LoadEnd,&Flash28_API_RunStart);

void Example_MemCopy(Uint16 *SourceAddr, Uint16*
SourceEndAddr, Uint16* DestAddr)
{while(SourceAddr < SourceEndAddr)
{ *DestAddr++ =*SourceAddr++; }return;}
```

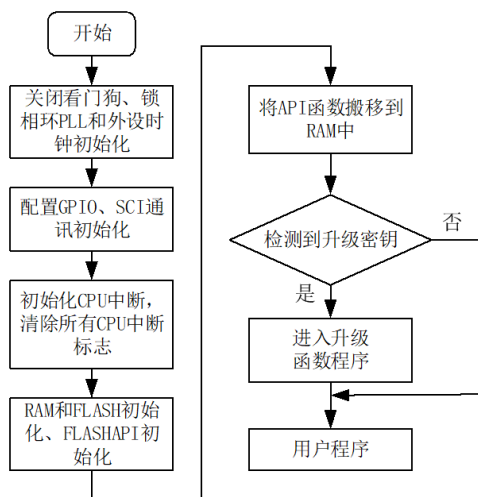


图 3: 升级程序流程图

3 双DSP嵌入式系统硬件设计

随着精度要求越来越高，为了提高控制精度，需要大量算法，需要一段时间进行计算，可能出现采样周期内算不完的情况，因此很多嵌入式控制系统需要更为强大的运算能力。本系统的硬件电路采用两个 DSP 并行处理的方式，实现了性能的拓展。

系统采集信号为 RS422 通信，通过芯片 MAX3490 将 RS422 通信转换为 RS232 信号，这些信号作为高速芯片 SC16C654 的输入，转换为数据线被双 DSP 接收，这些信号均可通过 CPLD 配置决定交给主控 DSP 处理还是辅助计算 DSP 处理。将双 DSP 的 Zone7 映射到 ST16C654 和 DA 芯片的外部存储空间，在 CPLD 中逻辑进行通道的选择。

双 DSP 均采用 TMS320F28335 芯片，在 CPLD 中进行功能分配使其能合理的并行处理运算，通过双口 RAM 型号 IDT70V28 进行双向通信，IDT70V28 可进行 16 位 64k 空间，最大延迟小于 25 纳秒的高速传输，最大限度地减少了双 DSP 之间通信的延迟时间。两个 DSP 都存在远程升级的问题，将两个 DSP 的 SCIA 通道作为远程升级的串口，连接到 CPLD 中。双 DSP 系统嵌入式系统框图如图 2 所示。

双 DSP 之间采用 CPLD 芯片 EPM2210F256。双 DSP 的 SCI 串口、各种控制信号以及外部的高速传感器信号都连接到 CPLD 中。CPLD 中进行时钟分频，产生控制信号进行芯片使能，逻辑控制，对通信通道进行灵活配置选择等。上位机和两个 DSP 进行通信，由上位机控制决定待烧写的 DSP 芯片，在 CPLD 中建立待升级 DSP 的 SCIA 通道。

4 远程升级软件设计

DSP 在线仿真时，程序直接从 page0 的地址 0X000000 中运行。然后跳转到主函数中运行。程序正常运行时，芯片上电后首先直接跳到中断向量表中 0x3FFFC0 处的 RESET 处执行，在这个地址下存放一个指令用于跳转至初始化引导函数 InitBoot，然后调用引导模式选择函数 SelectBootMode，用来检测配置为输入的 GPIO84~GPIO87 的引脚的状态，若四个引脚都为高电平则为 FLASH 启动方式，跳转到 FLASH 入口地址 0X33FFF6，此地址下存放了程序 codestart，codestart 为引导后重定向代码，在此程序中关闭看门狗，后跳转到 c_int00，_c_int00 的代码最终会调用主函数，后进入用户程序开始执行用户主循环程序。

.cmd 文件中主要包含 PAGE0、PAGE1 和 SECTIONS 三个部分，SECTION 为伪指令块，将数据段装载到相应的储存空间，PAGE0 和 PAGE1 为数据空间和程序空间，主要为指令 MEMORY，将装载的储存空间分配到 FLASH 区域中的具体位置。.cmd 为程序代码和数据分配响应的存储空间。

该文件为矢量跳转表文件，通常是汇编文件(.asm)形式，此文件需要准确的定位在程序的起始地址，其内容是汇编语句中的无条件跳转语句“LB”。

首先进行锁相环、时钟初始化、关闭看门狗。由于 API 函数包含具有软件延迟循环的时间关键型代码，这些代码必须执行以满足特定的计时要求。因此，在调用 Flash API 函数之前，设备必须以正确的 CPU 频率运行。如果设备的输入时钟丢失，PLL 将进入所谓的低电平模式，CPU 将以更低的频率计时，DSP 将被锁死。然后将预留的串口相应的 GPIO 管脚配置为 SCI 通讯模式，并初始化通讯波特率。初始化 CPU 中断并清除中断标志，API 必须保存全局中断之前的状态并在结束后恢复中断如果在擦除、程序在烧写过程中中断，FLASH 被锁死。因此中断使用的代码或数据不能存储在 flash 或 OTP 中。

将 API 函数库从 FLASH 区域搬运到 RAM 区域中。对 API 函数进行初始化，Flash_CallbackPtr 是全局函数指针，用于指定 API 操作时的函数，在 API 算法进行程序擦除、烧写和验证时都需要进行调用。所以首先进行 Flash_CallbackPtr 的初始化，将其定义为空。擦除 FLASH 将 FLASH 区域所有位置 1，擦除前要进行预处理，由于烧写过程需要对 GPIO 进行初始化，引脚将电平都拉高，所以进行预处理将关掉 PWM 等，防止 GPIO 电平拉高对系统产生影响。

最后进入程序主循环，检测 SCI 通讯时 DSP 收到的字符，当检测到密钥指令时正确时进入烧写函数 FlashIAP_UpdateKey，此函数通过指令 #pragma CODE_SECTION(FlashIAP_UpdateKey,"ramfuncs") 将烧写函数搬运到 SARAM 中运行。烧写函数进行。升级程序总体流程图如图 3 所示。

TI 公司软件 CCS 对整个工程文件进行编译和链接后格式文件 COFF，该数据格式复杂，不仅包含了以段的形式组织的代码和数据，而且还包含了文件头、符号表、段地址、初始化段入口等信息，程序中的代码和数据在 COFF 文件中是以段的形式存在，通过编译时配置不同的 CMD 文件，将这些段分配到不同的 DSP 地址空间。该文件格式的模块化结构与实际的内存存储区间不匹配，不能直接用来加载到 RAM 或写入 FLASH 中，需要将其转换成内存能识别的 HEX 数据格式^[16]。HEX 文件每段的代码可作如下表示：

: BBAAAAATTHHHH.....HHHCC

其中“:”为记录开始的标志；BB 为长度域，站一个字节长度，代表该段数据的数据字节数，通常数据字节数为 32 个字节。AAAA 为地址域代表该段第一个数据的地址，随后在此地址基础上增加以储存数据。TT 为类型域，表示该段数据的数据类型。类型域为 0X04，代表该行数据为扩展地址，类型域为 0X00 时代表该行数据为程序数据，类型域为 0X01 时代表该行数据为文件的结尾。HH 为数据域。CC 该段所有字节相加将模除 256 得到的余数取补码作为校验和，占一个字节长度。

程序中 Boot_LoaderFunc 函数为接收 HEX 文件函数，判断 SCI 通讯收数为“:”则代表接收开始，只要接收字节数不为零均开始接收，当数据类型为 0x00 时进行储存，数据类型为 04 代表扩展地址，接收数据类型为 01 时代表接收完毕。

5 实验验证

在双 DSP 伺服控制板上与 PC 机的上位机进行实验。通过上位机选择升级主控 DSP 的程序时将该 GPIO 拉高, CPLD 中逻辑进行判断当相应 GPIO 被拉高时, 主控 DSP 的 SCIA 通道和上位机进行交互。设置上位机波特率与程序中 SCI 配置相同。要升级时输入字符密钥“u”。将编译生成的 HEX 文件发送, 经过擦除、烧写和校验过程, 返回程序烧写成功。

本实验中主控 DSP 的用户程序为 GPIO60 每经过 0.1s 翻转一次, 返回烧写成功后重新上电运行程序, 通过示波器观测相应 GPIO 的输出波形, 可以看出反转周期为 0.1s 和预期一致, 证明烧写成功。实验测试升级过程, 烧写成功标志和测试结果分别如图 4 到图 6 所示。

6 结论

采用 TMS320F28335 芯片的嵌入式系统在工业控制上应用广泛。本文在双 DSP 嵌入式系统硬件上开发了一种基于 SCI 的程序升级方法, 解决了远距离或者复杂环境下程序无法升级的问题, 具有较大的实用性。本方案通过实验测试, 运行结果和用户程序一致, 表明程序烧写到 FLASH 区域, 验证了烧写方法的正确性。

参考文献

- [1] 张卿杰. 手把手教你学 DSP[M]. 北京航空航天大学出版社, 2015.
- [2] 司栋森, 李增智. 基于双 DSP 的高速数据采集控制器的设计[J]. 计算机工程, 2006(22).
- [3] 朱华征, 范大鵬, 马东奎. 基于双 DSP 的导引头数字控制系统设计[J]. 光电工程, 2009, 36(6).
- [4] 刘浩, 李荣冰, 刘建业. 基于串口通信的 DSP 在线烧写技术研究[J]. 电子测量技术, 2017(7).
- [5] 李静, 张树团. TMS320F2812 片内 Flash 在线烧写技术研究[J]. 国外电子元器件, 2008, 16(10): 37-38.
- [6] 何艳丽. TMS320F2812 在线编程中的文件格式转换[J]. 煤炭技术, 031(2): 220-222.
- [7] 高世皓. 利用 HEX 文件实现 TMS320F28335 的程序升级方法[J]. 单片机与嵌入式系统应用, 2014(7).
- [8] 罗秋凤, 叶慧, 李勇. DSP28335 嵌入式系统的 SCI 在线编程方法实现[J]. 河北科技大学报, 2013, 34(4): 318-324.
- [9] 郭浩田, 夏鯤, 李正荣. 基于 SCI 通信与双核系统的 DSP 在线编程技术[J]. 电子测量技术, 42(02): 109-113.
- [10] 王尧, 李荣冰, 孙永荣. DSP 在线升级容错性能的研究[J]. 电子测量技术, 2017(11): 193-196.
- [11] 何宾, 朱红林. 基于 GPRS 的 DSP 程序远程升级的研究与实现[J]. 计算机工程与设计, 2010(15): 58-60+118.
- [12] 罗显光, 朱跃欧, 李振. 基于 CAN 总线的悬浮控制器在线程序优化的实现[J]. 电力机车与城轨车辆, 2017(03).
- [13] 陈晓聪, 游林儒, 黄招彬. 基于 CAN 通讯便携式 DSP 程序下载器的设计[J]. 自动化与仪表, 2013(9).
- [14] 杨达亮, 陈军灵, 张力成. DSP 系统软件在线升级机制的研究和实现[J]. 电测与仪表, 2014, 043(001): 60-62, 24.
- [15] 邓中亮, 孙静. 嵌入式设备在线系统升级的设计与实现[J]. 计算机工程与设计, 2009(13): 51-53+56.
- [16] 徐鲁花, 董小卫, 张浩. 嵌入式开发系统编程文件格式解析

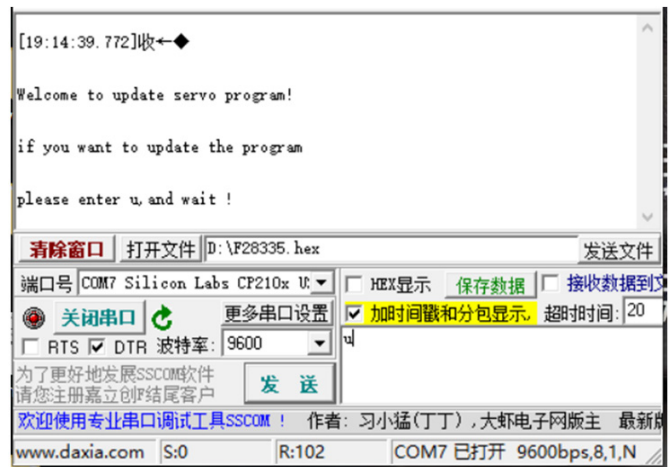


图 4: 上位机确认升级

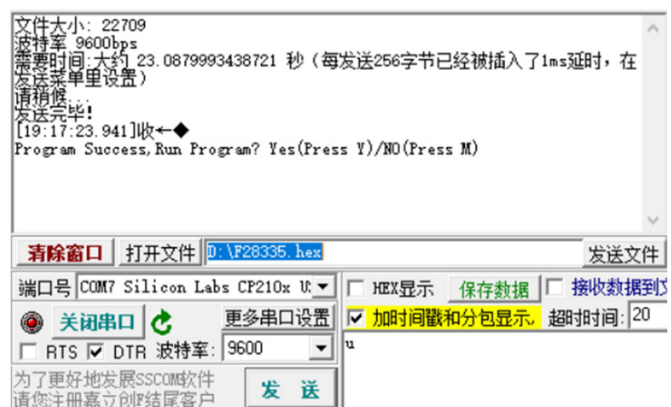


图 5: 上位机烧写成功

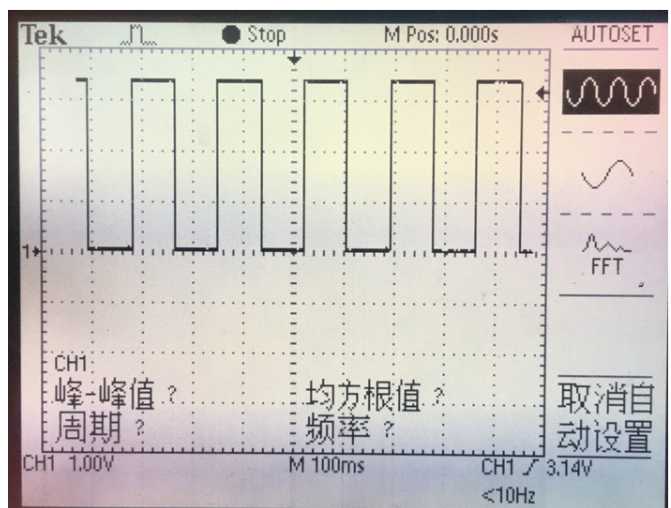


图 6: 程序翻转的 GPIO 测试波形

[J]. 单片机与嵌入式系统应用, 2011, 11(12): 4-7.

作者简介

王璐(1992-), 男, 黑龙江省牡丹江市人。硕士研究生。实习研究员。研究方向为伺服控制与嵌入式。