

圆梦小车升智商啦！

一、背景

圆梦小车在经历了“强身健体”、降低自身功耗、完善轨迹采样器安装结构一系列改进后，终于迎来了“大脑”的升级，从 8 位升至 32 位，从 51 跨越到 ARM！

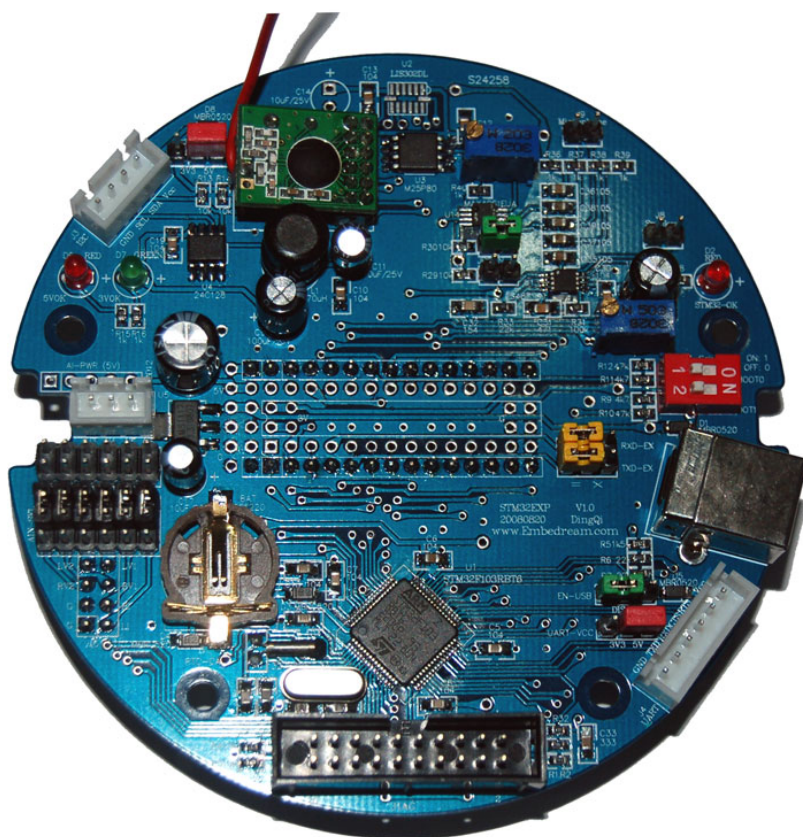


图 1 使用 STM32F103RBT6 的扩展板

圆梦小车的设计初衷就是想为想学习单片机、欲在嵌入式领域发展的大学生提供一个实践平台，帮助他们消化所学的理论，掌握所学的知识。所以在设计时就考虑了如何满足学习内容提升的需求，最大限度的发挥小车的作用，尽量使学生们从“口中省下”的钱花得值。

那个看似“多余”的扩展板及两排插针，正是为此而设计的。

二、“大脑”的选择

依我个人的经验，学习单片机应该从简单的开始，这样便于理解其工作原理；而且最好选择经典架构的，Intel 的 51 系列就是之一，其技术公开后，基于 51 内核的芯片层出不穷，足以证明其经典。而且国内大学授课多数还是以 51 架构为主。

所以开始选择了 51 系列 MCU，但为了让学生能接触到新东西，如 PCA，选择了改进型的 51，这样便于向新型单片机过渡。

实际上，很多单片机在描述自己的改进、提高时，都会用经典 51 作为参照系，所以掌握 51 系列单片机对于理解新型单片机有很大帮助。

但只学 51 够吗？ 肯定不行！

因为从学习角度看，做嵌入式控制，一定要掌握实时操作系统，否则会制约自己的发展。但在 51 这个级别上使用操作系统，有些勉强；必须升级到 16 位、32 位的单片机上才匹配。

由于个人精力有限，圆梦小车初始时只是提供了将 51 换为 Mega128 的可能，因为 128 的内存较大，可以跑一些精炼的操作系统。

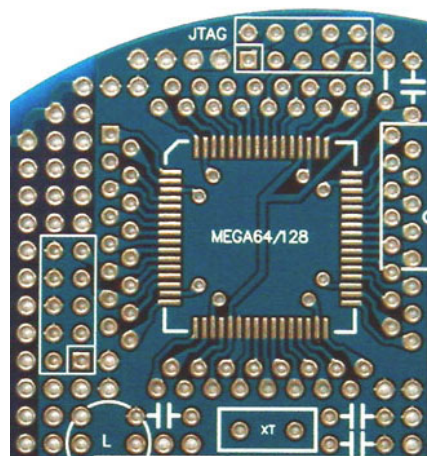


图 2 目前扩展板上 Mega128 的位置

但毕竟还是 8 位机，其工作模式与 16 位、32 位有些差别。所以没有花时间去实施，只是为有兴趣者提供了硬件上的方便。

由于 ARM 的强大攻势，使之几乎成了嵌入式领域的 x86，所以我也决定用 ARM 系列作为圆梦小车升级用的“大脑”。

ARM 是个十分丰富的系列，有针对不同应用领域的产品，如 ARM7、ARM9 到 ARM11 等。但早期的 ARM 似乎是针对便携式设备而设计的，典型的应用是 PDA、手机等。这类应用与嵌入式控制不同，它们强调的是“用尽量低的功耗实现尽量高的性能”；在内部结构上偏重于多媒体处理功能而非控制类的应用，如“位控制”在工业控制领域是常见的，而 PDA 上几乎不需要。

由于便携设备主要以电池为电源，抗干扰能力也不是主要考虑因素，但对于控制用 MCU 则是致命的指标！

多数单片机应用领域是工业控制，圆梦小车也是围绕此目的而设计的。所以在初期选择 ARM 时很困惑，似乎在选择依据上就不充足！？

好在近期 ARM 推出了面向控制应用的 Cortex-M3 系列，而且 ST 公司用其设计了一系列十分适合于控制的 MCU —— STM32 系列，消除了圆梦小车选择“大脑”的困惑！

之所以选择 ST 公司的 STM32 系列，主要是看中了他们为 STM32 所提供的完善技术支持，特别是完整的库函数，使用者几乎不必“扒着”手册逐个研究寄存器的配置。功能那么丰富的 Cortex-M3 内核，很难靠传统模式对付。

有人开始就选择 ARM 上手，我认为不妥，这就像上楼，如果台阶高到一定程度，就成了墙！而且由于 ARM 的结构过于复杂，考虑了各种应用需求，初学者根本无法理解其必要性和带来的好处；例如 I/O 口的改进，没有使用过 51，感受过那种准双向口的好处和局限，很难体会 ARM 的 I/O 口为何设计出那么多变化。

三、换“脑”方式介绍

圆梦小车设计时就考虑了换“脑”需求，所以采用了主控板、扩展板双层结构，将主控板上 MCU 的所有引脚都通过插针引到扩展板上，这样，如果主控板上的 MCU 去掉（或者使其

休眠），主控板上的外围资源就可以用扩展板上的 MCU 控制，十分方便。

关键是，在提升 MCU 时，可以参照原来的 MCU 控制模式，比较其中的异同，由此及彼，大大降低了理解新的 MCU 的难度，因为控制对象相同，所以很容易比较 MCU 升级后所带来的改变，以及不同 MCU 所适合的应用。

嵌入式控制领域与 PC 不同，它强调的是适合而非先进，所以通过同样的控制很容易比较出不同 MCU 的特质，从而为日后设计产品储备选择的依据。

由于新的 MCU 放置在扩展板上，有足够的空间扩展其特有的功能，从而使学习者涉猎到更多的知识，也使小车具备了更强的功能。

在此次 STM32 设计中，就针对其特征设计了 I2C 接口和存储器、SPI 接口和存储器、RTC（实时时钟）、USB 口、音频输入、音频输出等应用，使学习者能从中学到更多的知识，有更丰富的体验。还增加了三轴加速度传感器，我想它会使小车的控制更丰富。

四、新的“大脑”—— ARM 扩展板简介

从设计宗旨上看，这块扩展板主要功能类似于市场上已有的评估板，只是基于小车的资源做了量体裁衣的改变。保留了评估板的主要功能，但放弃了人机界面——显示、按键及摇杆等。还是那个原因：小车是要跑的，没有机会去看它、触摸它！

在工作模式上，考虑支持两种模式：

其一、完全替代原 STC12LE5412AD，对小车进行控制。

其二、作为上位机，通过 UART 控制 STC12LE5412AD，实现两级控制。并不是小车需要如此，而是这种模式在工业控制中常见，所以学习者应该尝试之，以积累相应的经验。

ARM 扩展板由以下几部分构成：

4.1、核心 STM32 F103RBT6

小车选择的是 64 脚的 STM32F103RBT6，考虑到这是一个兼顾上下的规格，64 脚封装覆盖了 STM32 的全系列，也就是说可以不改变 PCB，随意替换为这一系列的任何芯片。

选择 F103 系列是为了能够学习 USB，同时有足够的内存尝试 RTOS（128kFlashROM，20kRAM）。

主频选择 8MHZ，这是其推荐的，内部有灵活的 PLL，可以方便的改变时钟速度，从而兼顾功耗和性能的需求。

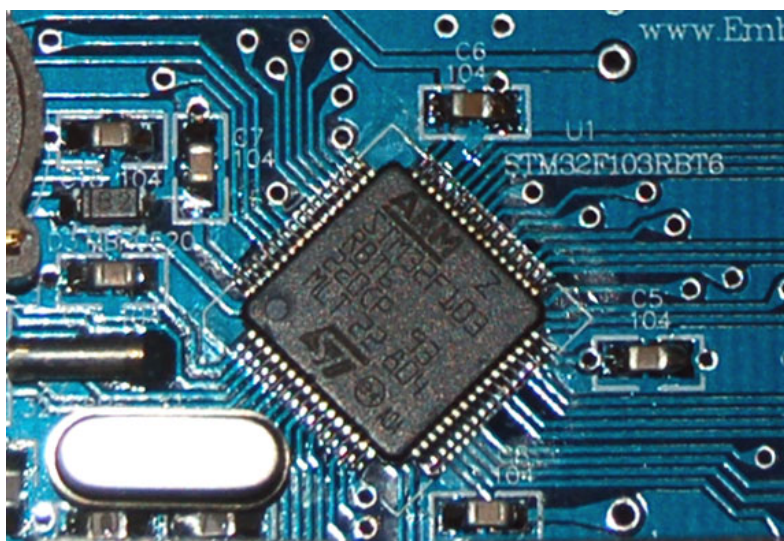


图 3 STM32 扩展板的核心 —— MCU

4.2、JTAG 口

设计了全引脚的 JTAG 口，以便于使用各种标准的 JTAG 仿真器。

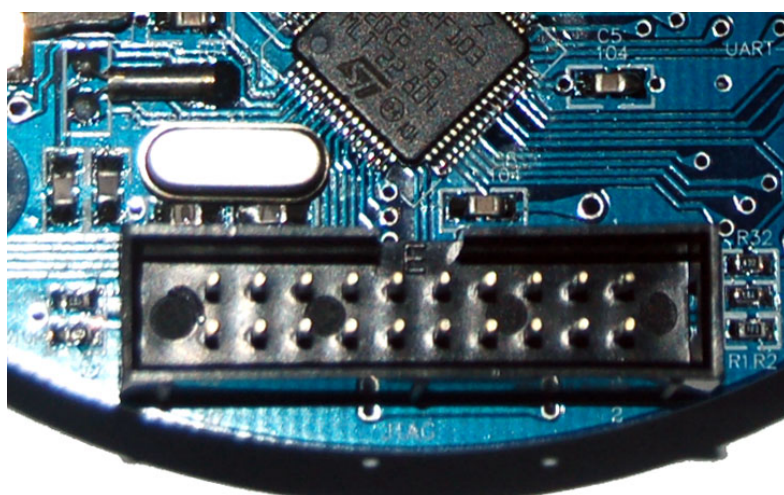


图 4 STM32 的 JTAG 口

ARM 不同于 51，太复杂了，需要借助仿真手段消化之。

我所使用的是南京万利公司的 ST-LinkII，支持 STM32F 系列，配有 IAR 的 IDE 环境，可免费编写 20k 代码，对于学习基本够用。



图 5 JTAG 仿真器和 IDE 环境

4.3、方便的 BOOT 选择

因为 STM32 提供了灵活的 BOOT 选择，所以在 PCB 设计上考虑了这个功能，使用者可以方便的体验其 BOOT 的功能。



图 6 STM32 的 BOOT 选择

4.4、RTC (实时时钟)

因为 STM32F 芯片内部设置了实时时钟功能，从工业控制应用角度看，也是比较常用的功能。虽说小车似乎用不上，但为了学习者考虑，设计了此功能，并配置了后备电池，也许学习

者能有出彩的应用：P

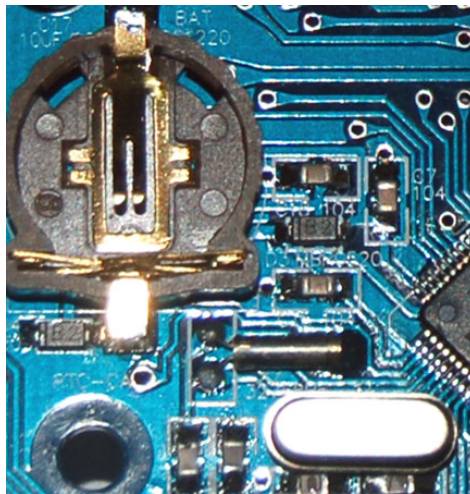


图 7 STM32 的实时时钟

4.5、模拟输入通道

STM32F 一大特征就是有高速的 12 位 AD，所以必须能给其展示的机会，正好圆梦小车二代增加了 4 路电机电势检测和 2 路电机电流检测，原 STC12LE5412AD 如果要采集轨迹，就没有足够的模拟输入用于电机检测了，而 STM32 的模拟输入足够应付。

为了能方便使用者接入自己需要的模拟信号，还提供了 6 个模拟输入口，并设计了跳线，可方便的在外接信号和电机检测信号之间切换。

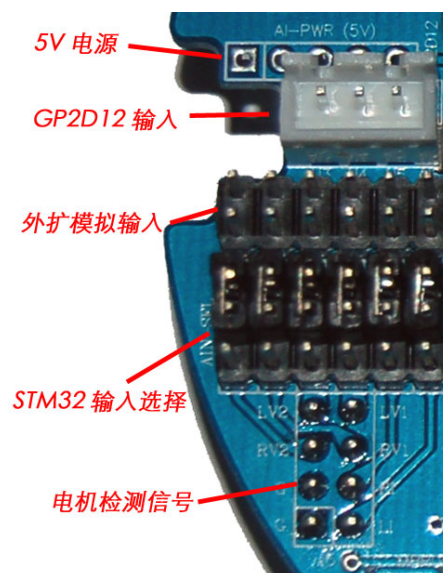


图 8 STM32 的模拟输入

从照片中还看到一个 3 芯白色插座，那是为了方便 GP2D12 输入的。

同时在白色插座上方还有一排焊盘，可以提供 5V 电源，如果外部模拟输入需要供电，利用此接口可方便的实现。

4.6 、I2C 接口和存贮器

I2C 总线在控制中经常应用，所以也是嵌入式控制常见的学习内容，小车上设计了一片 24C128 存贮器（因没买到，焊的是 24C256）。

由于 I2C 是靠指令区分设备的，扩充容易，所以设计了 I2C 接口，并且设计了 3V、5V 供电切换，以适应不同的芯片需求。

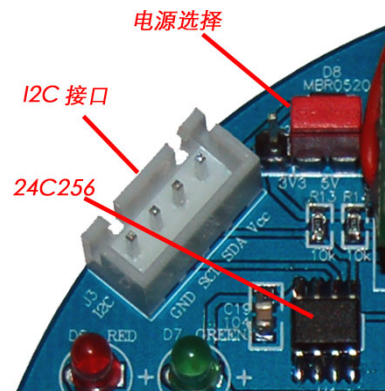


图 9 STM32 的 I2C 接口

4.7、SPI 总线接口和存贮器

SPI 总线和 I2C 类似，也是控制应用中不可或缺的一种，作为学习者必须掌握。

此处设计了 3 个 SPI 设备：

M25P80 存贮器 —— 考虑 M25 系列是典型的串行 Flash，而且 ST 也提供了例子，所以设计了，作为小车上的应用还未考虑好，不过因为设计了语音输入和输出，所以对于语音部分的处理可能有用。

RFM12B 无线模块 —— 这是小车必须的，在 ARM 级别学习上，不能再那种经串口转换的模式了，应该由 MCU 直接控制，实现无线通讯。

LIS302DL 三轴加速度传感器 —— 这是为小车新增的，期望能通过它控制小车走直线，

以及控制小车的速度，有了它，控制小车的素材、可以发挥的空间更大了。

不过由于该器件难以获得，正在向 ST 求助，目前还是虚位以待 ☹

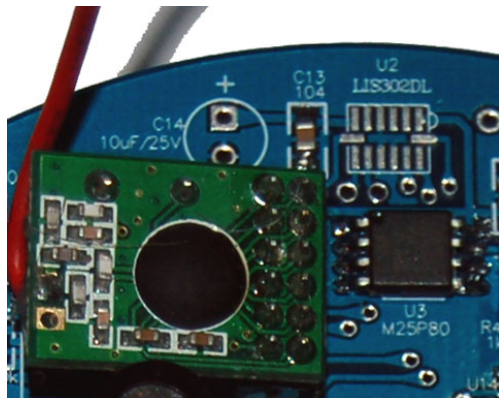


图 10 STM32 的 SPI 接口

SPI 接口设计时考虑了与原 MCU 的兼顾，因 STC12LE5412AD 也有 SPI 接口，所以将两者并联，学习者可以分别尝试一下，看看其中差异何在？ARM 到底好在哪里？

4.8、音频输入、输出

大概是为了展示 STM32F 的 AD 速度，所以在其评估板 STM3210B-EVAL 上设计了音频输入和输出功能，我觉得小车如果也能听话、说话倒也不错，即使不懂人语，能相互间用机器语交流也一定有趣，或者就用摩尔斯码。

所以就完全按照 STM3210B-EVAL 的音频部分设计了，只是没有将话筒和喇叭接上，主要是为了留给学习者自己发挥的空间。用户可参照 ST 的示例尝试之。

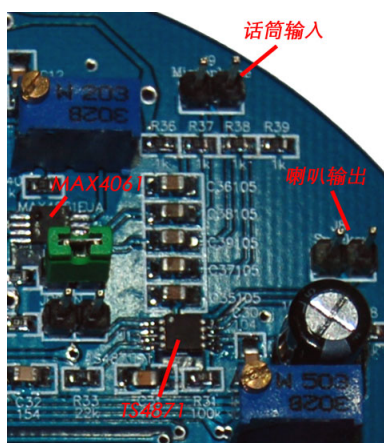


图 11 STM32 的音频输入、输出

4.9、UART 口、USB 口

STM32F 有丰富的串口，所以我设计时将一个用于和原 MCU 交互，另一个 UART 作为外扩，以便扩展一些通过 UART 的连接的设备，如蓝牙模块。电源也提供 3V 和 5V 选择。

与原 MCU 交互的 UART 口设计了收、发线交叉选择，如选择取代原 MCU 工作模式，则收发线不交换，原主控板的 UART 口和 STM32 相连。

如采用上下位机工作模式，则选择交叉模式，这样两个 MCU 就可以相互通讯了。

扩展 UART 口除将 TXD、RXD 引出外，同时引出了 RTS、CTS，为硬件握手提供了可能。如不需要，可作为与外设连接的控制信号，很多外设都需要一个中断和唤醒信号，而 STM32 的 I/O 口配置十分灵活，任意一个 I/O 口都可以配置为中断输入，且可以选择触发边沿，这就为硬件设计提供了极大方便，不用再在设计时考虑是否需要中断、是否需要反向等因素了。

USB 口作为 RS232 口替代者，学习者有能力还是应该掌握，所以设计了 USB 口。

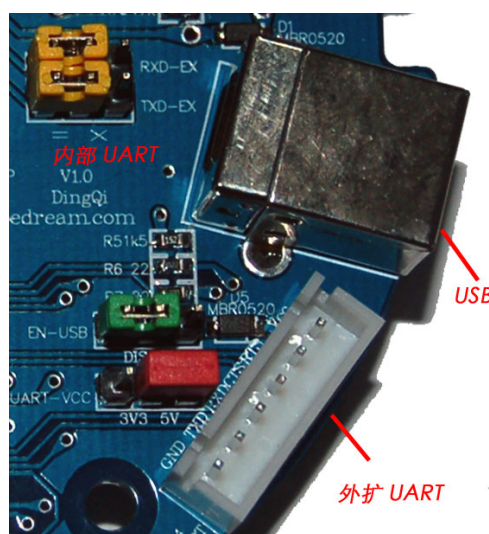


图 12 STM32 的 UART 及 USB

至此，小车 STM32 扩展板所提供的功能介绍完了，我觉得基本将 STM32 的资源用尽，有些是在软件上体现的，如 DMA。

从硬件角度，STM32 资源中，只有 SD 卡读写在小车设计中未考虑，因为觉得在控制领域使用 SD 卡的机会较少。也许是我孤陋寡闻吧：D

五、编程小试

为了测试 PCB 样板，必须编写程序。

好在略经挫折就搞定了 JTAG 口，IAR 的 Workbench 也通过阅读手册及模仿万利的示例程序勉强启用了。

立刻开始着手将原来的 Step 系列程序移植，因为在 STM32 扩展板上，我同样设计了工作指示灯，所以基本可以照搬。

经过努力，花了大约一周时间，我将 Step 1 - 3 的程序移植完成了，启用了 I/O 口、定时器及中断、UART 口及中断。在这个过程中也逐步消化了 IAR 环境的设置，以及 STM32 库的使用。ST 公司的例子很有帮助，基本都可以找到和你需要解决的问题相对应的示例（见我移植程序中的注释）。

具体细节见附件。

我只想大概说说这几天编程的感受：

其一、不太像单片机编程，更接近我在 VC 下编程的感觉，只是没有微软那样完备的函数自动弹出机制，需要你自己查阅库函数说明，参照例子使用。

其二、几乎不用接触汇编，编程至今，我还不知道 ARM 的汇编语句是什么格式呢，一条汇编指令也没有看过。

其三、不研究 STM32 的用户手册，搞清寄存器有何作用，也能将硬件启用。但如果能够略知一二，对快速理解库函数和排除故障有益。我在启用 I/O 口中断时就遇到麻烦，被迫研究了一下寄存器，并且借助 IAR 的 Debug 功能仔细观察了寄存器的变化。因为开始没有找到对应的例子，后来搞定后才发现在 EXTI 类中有 ☺

其四、用 C 语言写程序，基本不用考虑在什么单片机上运行。特别是从小内存单片机向大内存转换时，根本没有障碍。所以还是那句话 —— 尽量使用 C 语言。读者可以比较一下附件中的程序，看看移植后除硬件初始化外，其它有无改变？

其五、不懂英文可能确有困难，因为库函数说明、例子程序的注释都是英文的。如果你要

使用比较先进的芯片，估计看原文资料是无法回避的！

六、结语

虽说从上面组成介绍看，STM32 扩展板的功能和评估板相差无几，但实则不然。

因为评估板只有抽象的能力展示，没有需求。

小车则不同，STM32 扩展板只是“大脑”，它依附于小车这个“躯体”，这个“躯体”不是抽象的，它会产生出很多具体的需求要“大脑”解决，所以使用者很容易理解“大脑”为何设置了那么多功能？能干什么？

而评估板则不同，它是先抽象的展示出“大脑”的功能，再通过文字、语言告诉学习者，这些功能可以满足哪些需求？往往由于学习者常识的差异，理解会千差万别，也许根本没有理解。

如果为了具体需求而寻找出“大脑”的相应功能，并使用之，绝无可能理解错，因为那样将无法需求。

就像教孩子用水果刀：

方式一是先给孩子演示水果刀可以切断东西，然后告诉他可以用来削水果皮。

方式二是直接拿一个梨，先让他尝尝连皮吃的感受，再给他一把水果刀，教他如何用刀削皮。

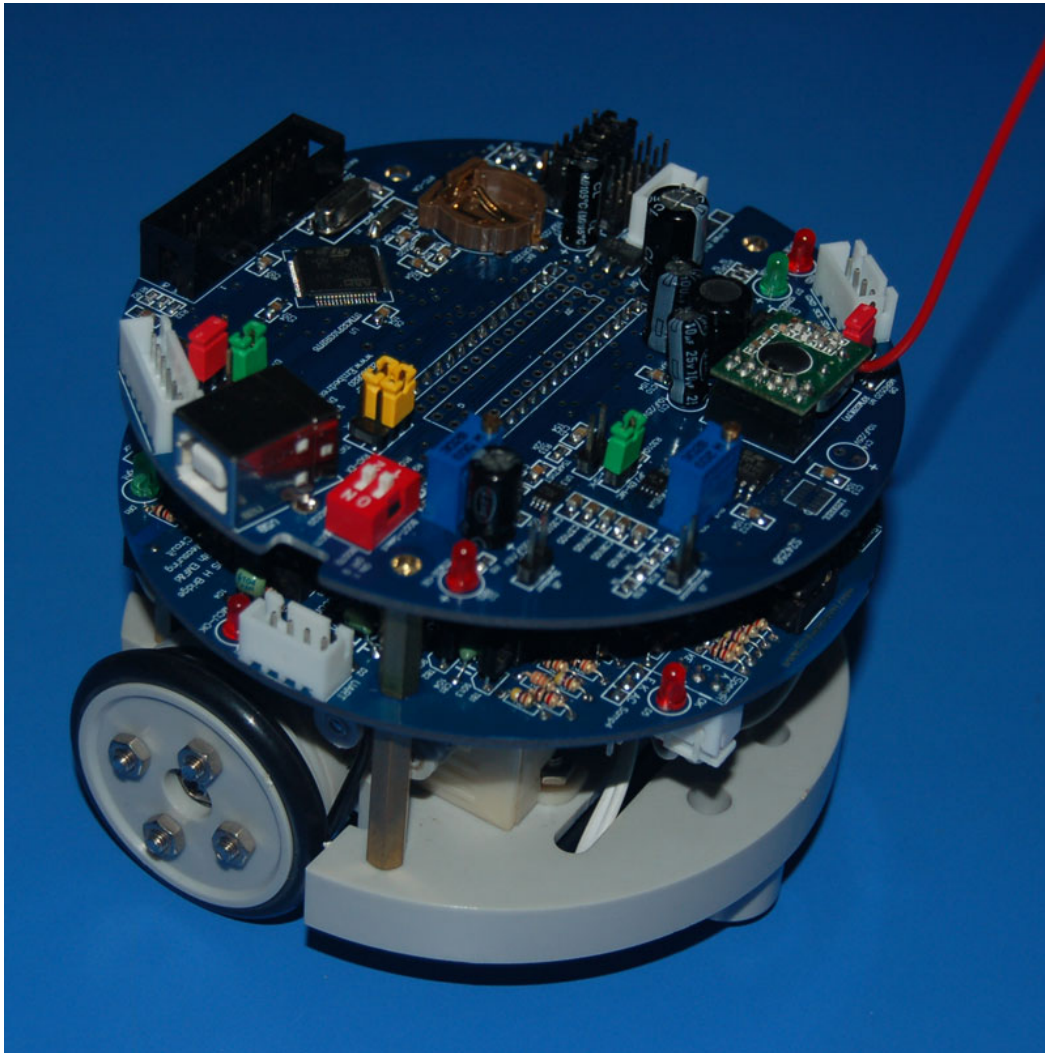
这两种方式那个更容易让孩子掌握？

我想肯定是后一种，而且孩子在使用中还可以弄清什么样的刀合适？怎么用合适？

这正是现在嵌入式控制教学的痼疾所在。老师往往是先罗列所教单片机的功能，并且一一分析，然后根据自己的常识假想一些应用，而且通常是一些十分勉强的应用。学生本来就对所学的东西模模糊糊，再弄个根本没听说过的应用，更是如坠云雾。

所以不要小看小车带来的需求，它会引导学习者准确的理解 MCU 所设计的功能。只有在准确理解的基础上，学生才能举一反三，将这个功能应用于自己日后的需求中。

装配后的小车如下：



由于 STM32 涉及的功能较多，目前还处在 PCB 测试阶段，我会尽快完成之，使圆梦小车的用户可以更上一层楼。

但愿它能够帮助使用者圆其嵌入之梦！

南京嵌入之梦工作室

2008 年 9 月 25 日星期四