EduCake 的入门介绍和 Digital 脚位功能的使用

一、 规格介绍与 IO 脚位

在日常生活中,各式各样的电动玩具、电器、遥控、或工厂里生产线的控制、 各种机器的控制,都存在着单芯片。但对于一般非电子相关科系的人来说,学习使 用单芯片有着很大的困难,常常需要熟悉各种电子零件的功能、电路的架构、复杂 的低阶程序撰写等等,得要不少时间的工夫养成,初学者常常接触后不久就放弃了。

这状况终于在数年前获得较好的解决方式,由意大利工程师所开发出 Arduino 开放式架构单晶开发平台,依旧属于单芯片,但不同的是 Arduino 所内含与设计的 开发平台非常适合初学者使用,不论是语法精简、接口和善、或是中文书的辅助、 还是网络资源众多、相关零组件或功能模块价格便宜、易取得等皆是所具备之多项 优点,更重要的是电路和资源的完全开放,促使这几年间广泛的、大量的被运用在 教学和专题上使用。

由台湾厂商独立开发设计、自行制造的 86Duino 系列控制板基于这样的条件下 问世了。这个由台湾厂商独立研发制造的 86Duino 系列却拥有如下的惊人规格:

- ▶ 300MHz 的 32-位架构 x86 平台 CPU 作核心
- ▶ 内建 128MB DDR3 高速内存
- → 有 PC 等级的局域网络 LAN 接口
- ➢ USB 2.0 * 2
- Micro-SD
- > Open-Source Hardware
- Support DOS, Windows, Linux

▶ 提供 Arduino-完全兼容的开发 IDE 接口,这一项,使得 Arduino 的使用者不 用再花费时间学习新的东西,直接拿来用,原有 Arduino IDE 所附带的数十 个范例程序,全部都不用修改就可在 86Duino 上执行;许多 Arduino 自带 的函式库,在 86Duino 上也都有支援。

86Duino 系列里面的 EduCake 教育版本,直接和面包板结合,从外观清爽明 义,搭配喷沙金属外壳及白色的面包板,十分有质感,是一个非常适合用来实作和 教学用的微电脑控制板。



图 1-EduCake 立体图

图 2-EduCake 背面

图 3-EduCake 正面



图 4-EduCake 尺寸图

板子上可以清楚看到 Digital 0~13、Analog0~5、GND、5V、3.3V、RX/TX 等 Arduino Leonardo 完全一模一样的脚位(功能和程序写法也都相同,一个字都不 用改)。



- Digital0~20, 31,32, 42~44, 共26个 digital 可用来作为一般数字 IO 的使用, 电流量最大 16mA,都有限流保护,防止不当使用的故障,让初学者更可放心 大胆尝试各种应用功能。
- 2. Analog 0~5 做为模拟输入使用。
- 3. TTL 共有三组,分别是 RX/TX、RX2/TX2、RX3/TX3,对于各种通讯应用非常 方便
- 4. 脚位旁边有标示"~"符号者为支持 PWM 的脚位,此部分与 Arduino 相同,但 86Duino 所出的 EduCake 多了 Digital 13、31、32 等三个脚位也是支援 PWM。
- 板上多了 I² C 专用的 SCL/SDA 脚位,不用像 Arduino 328 要使用 I² C 就必须 牺牲 Analog 4、5 两脚,方便许多。

- EA0~1、EB0~1、EZ0~1 是专门 ENCODER 给 Motion Control 用,此功能
 Arduino 可就得要另外加控制板才可作到。
- 7. 5V 的脚位是直接 By Pass、3.3V 最大输出 800mA

想必原先 Arduino 玩家看了之后也会觉得很亲切,不止操作上相同,还额外增加许多功能,尚未使用过的玩家也可快速上手;在外观尺寸上大小适宜,刚好一只手可掌握;而金属外壳将相关电路包裹住,兼具保护及美观之功能。



二、 开发接口

说到 86Duino 开发接口即可十足可以感受到 DMP 开发团队的用心,画面除了颜色之外,与 Arduino 标准 IDE 接口一模一样,操作方式也当然相同了,真是使用者一大福音!!以下是正式版本的 IDE 画面,是不是感觉很熟悉呢?



图 6-开发接口与 Arduino 的接口十分相似

再来,看到 File-> Example 里面的默认范例都差不多

86Duind

www.86duino.com

New	Ctrl+N	. <u>o.</u>	
Dpen Sketchbook	Ctrl+O		
Examples		01.Basics	
Close	Curry	02.Digital	
Save	Ctrl+S	03.Analog	AnalogInOutSerial
Save As Upload Upload Using Programm	Ctrl+Shift+S Ctrl+U er Ctrl+Shift+U	04.Communication > 05.Control > 06.Sensors > 07.Display > 08.Strings > 09.USB > 10.StarterKit >	AnalogInput AnalogWriteMega Calibration
Page Setup Print	Ctrl+Shift+P Ctrl+P		Fading Smoothing
Preferences	Ctrl+Comma		
Quit	Ctrl+Q	ArduinoISP	
		EEPROM > Firmata > Liquid > SD > Servo > SPI >	

图 7-开发接口操作模式也很相近

若是需要相关的指令参考,<u>http://www.86Duino.com</u>网站中都有详细介绍, 基本上就是 htpp://Arduino.cc 官方网站里面的 reference 相同的东西,若是已经 看过的玩家可以直接跳过。当然,很多时候写程序都是需要参考,有很多的基本应 用范例、接线图等等可以采用,另外也可以当作指令速查字典使用,很方便。

不过 86Duino 开发 IDE 的档案目录结构和 Arduino 的结构是有落差的,但对 我们目前开发运用没有任何阻碍,若是有兴趣想要自己改函式库的读者可能得要先 重新熟悉一下,这部分后面会有更深入介绍。 程序写好以后要放到板子上的流程都相同,唯一要注意的是板子必须是 86Duino EduCake 这个选项,不要弄错了(如下图)。



图 8-开发接口 板子名称请选 86Duino EduCake

板子选择好以后,SerialPort的选择也要正确,笔者的号码是COM1。COM的号码可以随时在控制面板里面的系统-->设备管理器作任意修改。请注意这个画面右下方的文字:86Duino EduCake on COM11,意思是IDE 接口抓到板子是在COM11,但我已经修改成COM1,这部分使用前须要再三确认正确才行。

86Duind

www.86duino.com



图 9-开发接口 COM11 设置

选好 COM 以后,直接点下这个向右箭头就可以开始上传程序到 EduCake 板子上,开始跑啰。至于左方那个打勾符号是 Verify 程序的功能,笔者几乎从来不用的, 主要是,反正写好上传的时候就会执行检查、编译、上传的连续动作,有没有错也都会知道,不需要额外作这个动作。

86Duind

www.86duino.com



图 10-开发接口 开始写程序了!

三、 第一个程序

先来实作一个最简单的程序试试看。首先,准备一个 LED 和一个电阻,像下图 这样把线路接好。电阻不用也没关系,因为板子的脚位电流很小,即使电压超过电 阻需求也不会烧掉的,只是加电阻来保护 LED 是使用各种电路的好习惯,最好还是 继续维持。



图 11-将准备好的 LED 及电阻 接上 EduCake

请注意 LED 的正极接在 digital 脚位 3;然后开启 IDE 接口,输入以下程序:

86Duino www.86duino.com

```
void setup()
{
    pinMode(3, OUTPUT); // 設定腳位 3 為輸出模式
}
void loop()
{
    digitalWrite(3, HIGH); // 設定腳位 3 HIGH · LED 燈會亮起來
    delay(1000);
    digitalWrite(3,LOW); // 設定腳位 3 LOW · LED 燈會熄滅
    delay(1000);
    }
```

上传程序后就可以看到 LED 灯间隔一秒钟的时间在一闪一灭,循环不断,这个 范例是否和你写过的 Arduino 程序一模一样呢,简单吧。后续还可以修改这个程序, 使用 Digital 脚位的 PWM 功能,搭配 analogWrite 指令就可以控制灯光的慢慢变 亮或是慢慢变暗,或是加入稍微的随机数,来制造简单的烛光摇曳的效果,您会发 现,光是一个 LED 灯都可以玩出无数变化呢。

到目前为止,使用方式跟 Arduino 是完全一模一样的,程序也很简单,对于有 经验或是第一次玩这个的读者来说,应该能完全的上手了才对。

四、 第二个程序

第一个程序会动以后,我们就可以好好的来试验这块板子的功能了。把第一个程序延伸,多使用几个脚位来做 LED 的效果,不过因为考虑面包板的空间安排,此次使用八个 Digital 脚位来接 LED,接脚图如下:

首先, 接好八颗 LED 和电阻





换个方向看看



图 13-接好 8 个 LED 及电阻,换各角度看一下



考虑到后面需要用到按钮来作别的事情,也把按钮线路先接好

图 14-将其他线路也一并接好

接好电路以后,即可利用这个电路来实做很多种不同的东西,先来做个可以类 似跑马灯的程序,让八个 LED 灯连续不断的循环亮起,程序代码如下:

86Duino www.86duino.com



其中那行

```
pos ++;
```

if (pos>=8) pos =0;

其实是以下程序代码一样的东西,要如何写就看个人习惯了

最后那行 delay(200)的指令,也可以利用变量去变化他,来做出本来跑很快,然后渐渐变慢的效果等等。

五、 第三个程序

前面的程序能够动以后,就可以开始做更多的变化,接下来换利用 COM 的通 讯功能来跟计算机进行沟通。计算机端也需要对应的应用程序,这样就可通过这个 接口从计算机端来控制 EduCake。这个应用必须分两边写程序。

一边是在 EduCake 上面写以下这段程序:



86Duino www.86duino.com



digitalWrite(led[ch-49],HIGH);

ch-49 代表的意思是计算机里面的每一个字符(不管中文、数字、英文或是任何 其他国家的文字)都有对应的代码,数字 1~8 的 ASCII 对应号码分别是 49~57,即 可利用这个简单的减去 49 的计算,转换成 0~7,正好对应到上面设定的 Digital 脚 位的那个数组内容

好了以后,就可以利用 IDE 接口里面的 COM 监视器(Serial Monitor)来和计算 机通讯,输入 1~8 来控制对应的灯号亮起来,画面如下

86Duind

www.86duino.com



可在实心箭头处输入1~8的数字直接控制对应的LED灯亮起来。这个画面很常被使用在程序开发过程中,输出执行过程的讯息,用来作逻辑的侦错还满方便的; 而且也常用在通讯功能的测试部分。

不过要注意会占用 COM,所以若像后面会使用 VB/C#来写 UI 画面去控制 EduCake 的时候,要注意这个 Serial Monitor 就不能开,不然程序会有冲突。所以 若是这种情况,UI 执行前要记得关闭这个画面,否则有些严重的情况下,计算机得 重开机才能重新让 UI 抓到 COM

也可以用任何能写 COM 通讯的程序来和他通讯,我们这里使用 VS2008 版本 (2005~2013 版本都可以使用不会有错误)的 VB.net/C#来试试看,画面布置如下

Comm. Settin	igs		
COM COM	1 ▼ 連		
BPS 9600	 ■ 線 		
點亮	點亮	點亮	點亮
點亮	點亮	點亮	點亮



首先,设定好 COM Port 和 9600 的传输速率,按下联机按钮,就可以和 EduCake 作沟通。

这部分因为属于窗口程序,不在本文章讨论范围,就请自行观看附件的 VB 程 序了。若要有更多变化,也可以结合鼠标的滚轮来作往前滚动,LED 就依序亮起; 往后滚动 LED 就依序熄灭的应用。或是结合可变电阻来转动,电阻越大就量越多。 后面学会 Analog 的脚位功能后,还能搭配来测试电池的电量,电力越强就亮起越 多的灯...等等,应用的领域就端看您的想象能力到哪里啰。

有些人需要 C#的版本,可以到网络上搜寻"VB2C#"这样的关键词,就可以找到 一些转换的网页,程序代码很简单,转换不会有什么问题的。后续我们在项目中还 会结合手机的版本来做更多的应用。

不过这个程序有点瑕疵,灯号亮起来就不会熄灭了,这样变成若是按到已经按 过的号码,也不晓得到底是有没有送出讯息来控制灯号。这问题很好解决,可以把

修改成这样

<pre>{ digitalWrite(led[cf</pre>	ר-49],HIGH);				
digitalWrite(led[ch-49	9],HIGH);				
delay(1000);	-18-				
digitalWrite(led[ch-49],LOW);					
}					

www.86duino.com

让他亮一秒后就变暗,这样的就可以让整个程序很好的运作了。有人会问,那 干嘛不一次写好?

我们必须了解到,学习的过程中,不可能直接有完美答案的,很多时候是一个 功能先要求有答案,然后再慢慢的加入修正去让他稳定运作,加入新的功能让他慢 慢的完善,这整个测试的过程也才能体会到真正的程序和控制的乐趣哩。而且更重 要的是,往往一直修改的过程中,会一直体会到不一样的新写法,对功力的进步也 是有帮助的。

六、 第四个程序

了解了前面的基本功能后,就可以来写更完整实用的功能。不过因为这个章节

```
IING
int led[]={
 3,5,6,9,10,11,13,31};
// 因為要製作類似貪食蛇那種效果的燈號移動方式,
// 所以這裡使用上三個變數用來控制燈號的發亮位置和方向
int nowPos =2; // 開頭發亮的燈
int midPos=1; // 中間發亮的燈
int lastPos=0; //尾部發亮的燈
int dir=1; // 燈號發亮移動方向,1:小到大,-1:大到小
int spd = 20; // 移動速度
void setup() {
 Serial.begin(9600);
 for(int a=0;a<8;a++)
   pinMode(led[a],OUTPUT);
 randomSeed(analogRead(0)); // 一開始先來初始化亂數種子
}
void loop() {
                         -19-
 if (Serial.available())
     {
```

www.86duino.com

讨论 Digital,这里就来作一个类似电动玩具机台上的那种猜数字灯,一开始 LED 会 往某个方向一路亮过去又亮回来,速度越来越慢,慢到某种程度的时候,会随机停 在某一个 LED 发亮。这样可以用来制作像是电子骰子、吃角子老虎之类的应用,并 且和计算机 UI 作结合,用来控制他产生骰子号码 1~8 号(因为有八个 LED 灯,当然 读者也可以自己改成对应普通骰子的六个灯), Edu Cake 端的程序代码如下:



86Duino www.86duino.com



86Duino www.86duino.com



是依照 spd 变量来作 delay,这部分可以自行调整速度,或是改成使用 UI 来控制速度的变化,哪一种作法比较有趣就看读者自行发挥了。

在来就是,如果没有写 UI 或是不想用 Serial Monitor 来控制,想要直接按按 钮来控制他重新跑一次,则可利用前面已经接好的按钮(在 Digital 12),在执行的过 程中检查该按钮状态,若有按下就重新执行。修改后的程序代码如下(重复的部分就 用。。。省略了):

```
0 0 0
void setup() {
  0
 pinMode(12,INPUT); // 多這行,先設定 12 腳位為輸入
}
void loop() {
 int bb;
 bb=digitalRead(12); // 讀取 Pin 12 的狀態
 Serial.println(bb); // 順便印出來作除錯和其他用途
 if (bb==1) // 1 代表被按下
   run_again(); //呼叫重新執行的函數
}
void run_again() // 在主 LOOP 外面寫這一段,程式重新執行的設定用途
{
 spd = 20;
 midPos=nowPos-1;
 lastPos=nowPos-2;
 dir =1;
   }
```