

Notebook: billwang168 的記事本
Created: 2019/1/4 上午 11:51
URL: <https://www.itread01.com/content/1504365621.html>

巧妙的IO口按鍵掃描方法

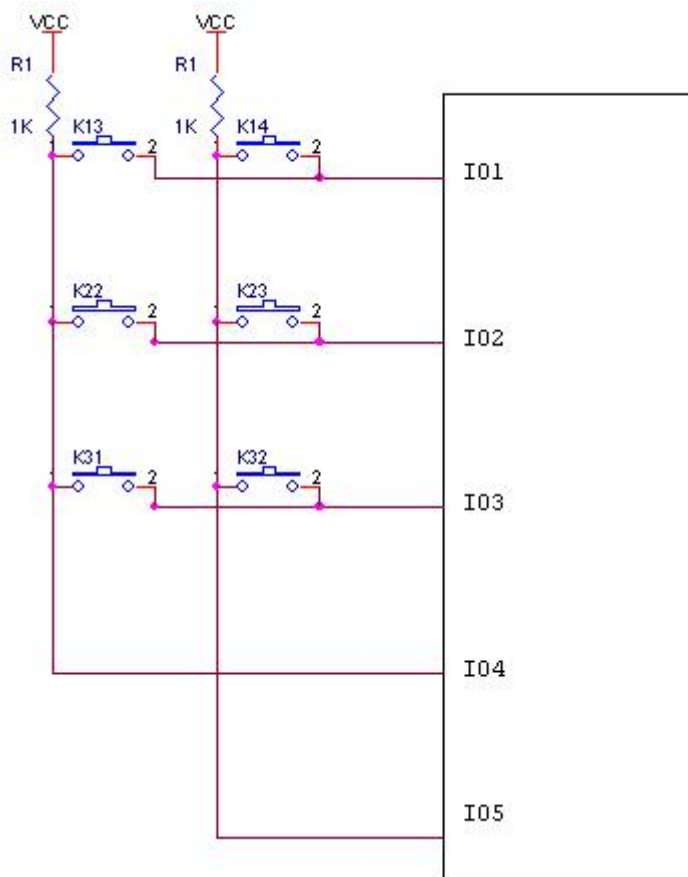
其他 · 發表 2017-09-02

添加 files 模塊 不能 zoom 二極管 2個 排列 什麼

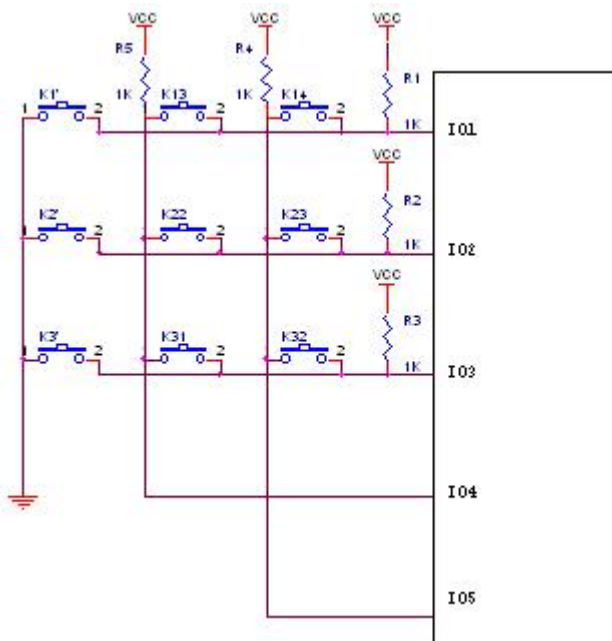
在做項目的時候，我們經常要用到比較多的按鍵，而且IO資源緊張，於是我們就想方設法地在別的模塊中節省IO口，好不容易擠出一兩個IO口，卻發現仍然不夠用，實在沒辦法了就添加一個IC來掃鍵。一個IC雖然價格不高，但對於大批量生產而且產品利潤低的廠家來說，這是一筆不菲的開支！

那，我們能不能想到比較好的掃鍵方法：用最少的IO口，掃最多的鍵？可以嗎？

舉個例：給出5個IO口，能掃多少鍵？有人說是 $2*3 = 6$ 個，如圖一：

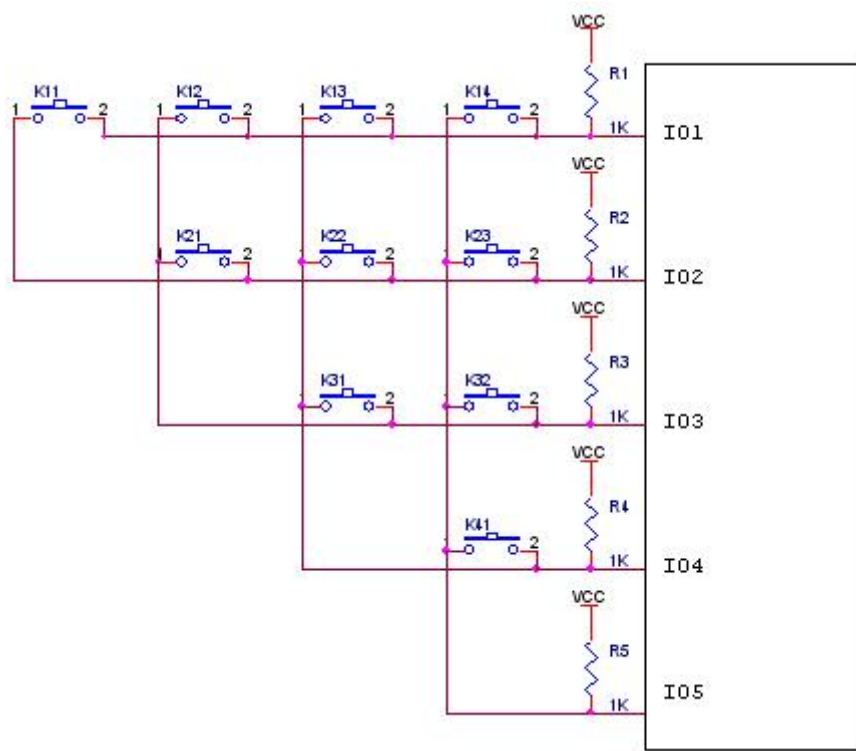


對，大部分技術參考書都這麼做，我們也經常這樣做：用3個IO口作行掃描，2個IO作列檢測（為方便描述，我們約定：設置某一IO口輸出為“0”一稱其為“掃某IO口”）。用行線輸出掃鍵碼，列線檢測是否有按鍵的查詢方法進行掃鍵。掃鍵流程：在行線依次輸出011，101，110掃鍵值，行線每輸出一個掃鍵值，列線檢測一次。當列線檢測到有按鍵時，結合輸出的掃鍵值可以判斷相應的按鍵。但是，5個IO真的只能掃6個鍵嗎？有人說可以掃9個，很聰明！利用行IO與地衍生3個鍵（要註意上拉電阻），如圖二：



掃鍵流程：先檢測3個行IO口，對K1'，K2'，K3' 進行掃鍵，之後如上述2*3掃鍵流程。5個IO口能掃9個鍵，夠厲害吧，足足比6個鍵多了1/2！

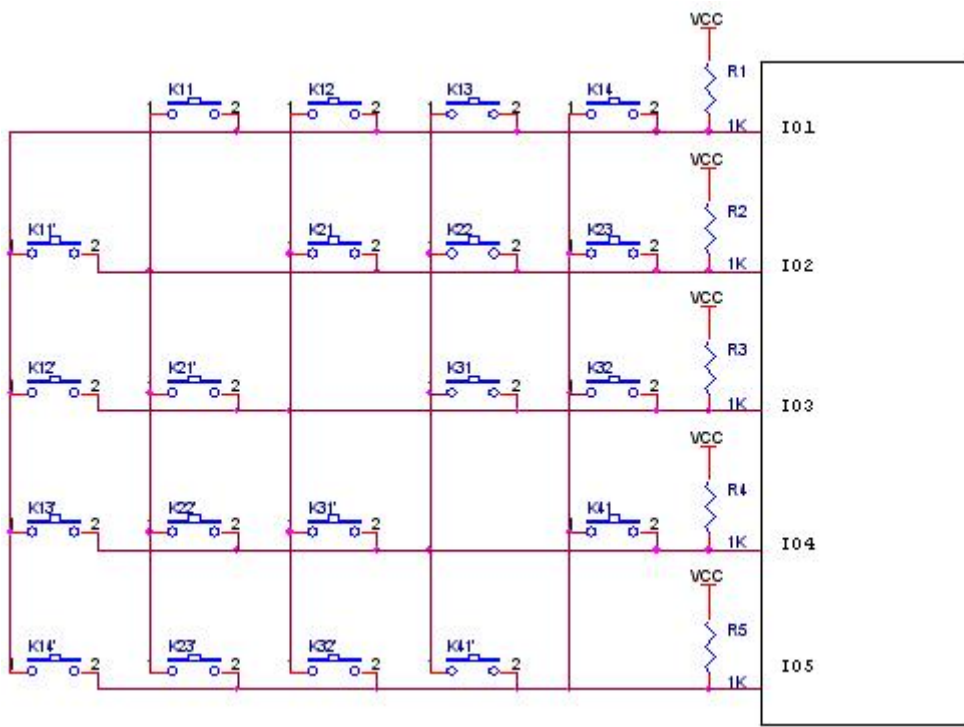
動動腦，還能不能再多掃幾個？就幾個？一個也行！好，再想一下，硬是被逼出來了！如圖三：



圖三

不多不少，正好10個鍵！這種掃鍵方式比較少見吧！漂亮！掃鍵流程：設IO1輸出為“0”，檢測IO2...IO5，若判斷有相應鍵按下，則可知有鍵；若無鍵，則繼續掃鍵：設IO2輸出為“0”，檢測IO3, IO4, IO5，判斷有無鍵按下，如此類推。這裏應注意：當掃某一IO口（輸出為“0”）時，不要去檢測已經掃過的IO口。如：此時設置IO2輸出為“0”，依次檢測IO3,IO4,IO5，但不要去檢測IO1，否則會出錯（為什麼，請思考）。

感覺怎麼樣？不錯吧！讓我們再看看圖三，好有成就感！看著，看著.....又看到了什麼？快！見圖四：

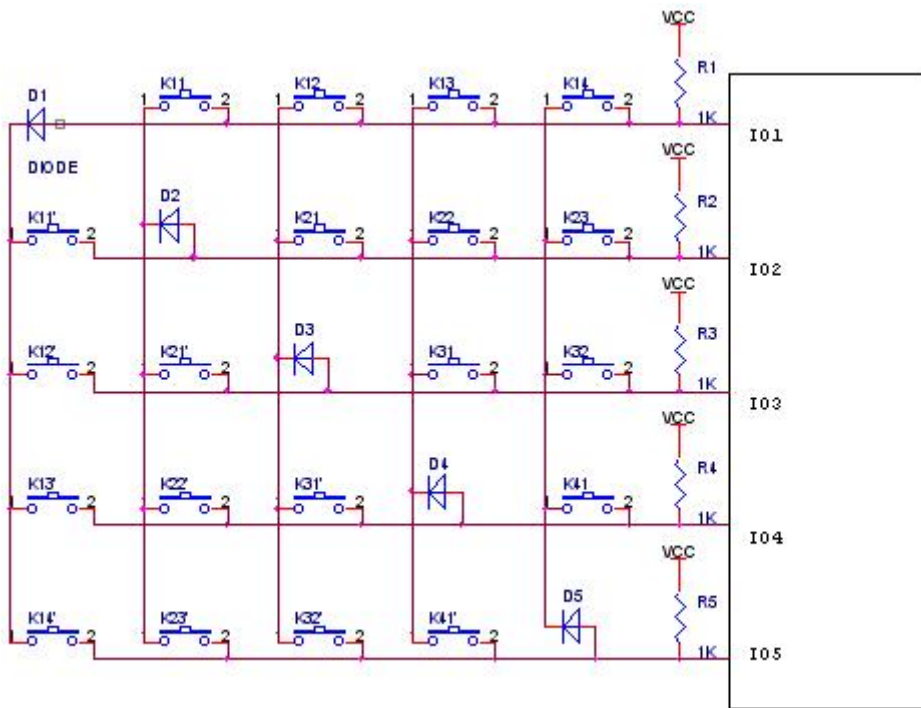


圖四

真強！被您看出20個鍵！多了一個對稱的三角形。可是，像這樣的排列能正確掃20個鍵嗎？回答是肯定的：不能！上下三角形相互對稱，其對稱掃出的鍵無法區別。有沒有注意到分析圖三時提到的注意點？（à “當掃某IO口時，不要去檢測已經掃過的IO口，否則會出錯”）

我們分析一下圖四：當IO1輸出“0”時，按下K11或K11’鍵都能被IO2檢測到，但IO2檢測卻無法區別K11和K11’鍵！同理，不管掃哪個IO口，都有兩個對稱的鍵不能區分。

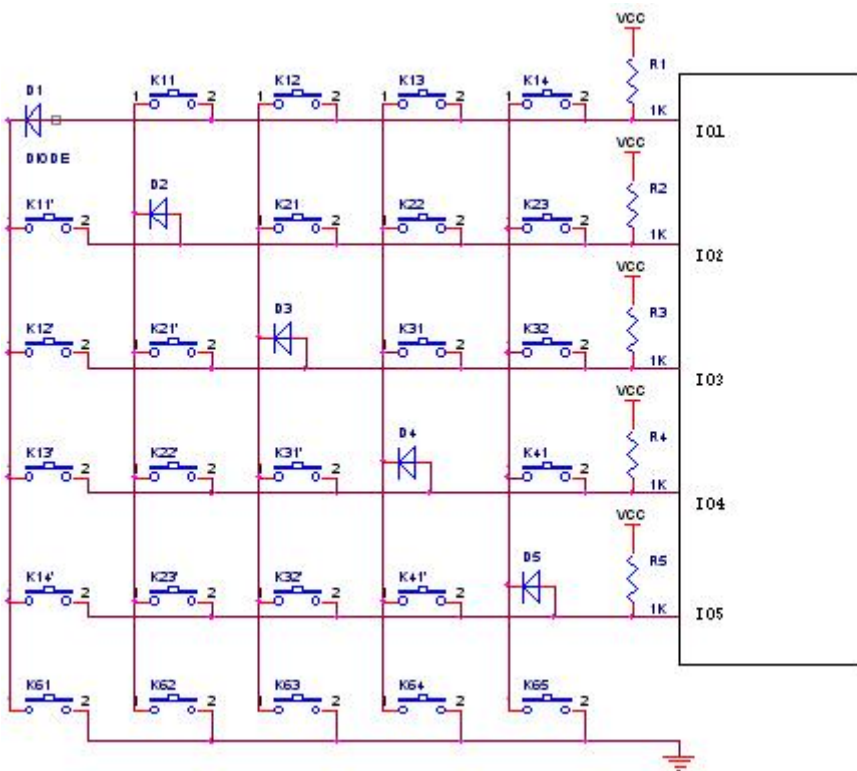
我們假想，如果能把對稱鍵區分開來，我們就能正常地去判斷按鍵。我們在思考：有沒有單向導通性器件？有！見圖五！



圖五

很巧妙的思路！利用二極管的單向導通性，區別兩個對稱鍵。掃鍵思路：對逐個IO口掃鍵，其他四個IO口可以分別檢測其所在的四個按鍵。這樣，就不會有分析圖三時提到的註意點。

夠酷吧！等等，大家先別滿足現狀，我們再看一下圖二，是不是有點啟發？對，我們再分析一下“用5個IO口對地衍生的5個鍵”。看圖六：



25個鍵！5個IO口掃出25個鍵！先別激動，我們再分析一下它的可行性，分析通過過才能真正使用。假設掃鍵流程：先掃對地的5個鍵，再如圖五掃鍵。先掃對地5個鍵，判斷沒有按鍵，接著對逐一對IO口進行掃鍵。但當對某一IO口掃鍵時，如果有對地的鍵按下，這時有可能會誤判按鍵，因為對地鍵比其他鍵有更高的響應優先級。例如：掃IO1，IO1輸出“0”，恰好此時K62按下，IO2檢測到有按鍵，那就不能判斷是K11還是K62。我們可以在程序上避免這種按鍵誤判：若IO2檢測到有按鍵，那下一步就去判斷是否有對地鍵按下，如果沒有，那就可以正確地判斷是K11了。

我們小結掃鍵個數S：

$$S = (N-1)*N + N \text{ — 啟用二極管}$$

$$S = (N-1)*N / 2 + N \text{ — 省掉二極管}$$

巧妙的IO口按鍵掃描方法