

Upboard Lab01

- Upboard Lab01
 - Some Useful Functions
 - mraa_gpio_context mraa_gpio_init (int pin).
 - mraa_result_t mraa_gpio_dir (mraa_gpio_context dev, mraa_gpio_dir_t dir).
 - mraa_result_t mraa_gpio_write (mraa_gpio_context dev, int value).
 - int mraa_gpio_read (mraa_gpio_context dev).
 - 擴充GPIO腳位數量
 - 74HC165晶片
 - 1. 介紹
 - 2. 原理
 - 3. 74HC165晶片 腳位介紹
 - 4. 74HC165晶片 操作步驟
 - Hands-on Practicing
 - Visual Studio Code
 - 作業繳交

Some Useful Functions

mraa_gpio_context mraa_gpio_init (int pin)

功能：初始化 GPIO 針腳。

參數：(int) gpio pin

回傳：gpio context or NULL

mraa_result_t mraa_gpio_dir (mraa_gpio_context dev, mraa_gpio_dir_t dir)

功能：設定 GPIO 針腳輸出 / 入方向。

參數：

a. (mraa_gpio_context) dev：代表 GPIO 針腳狀態內容。

b. (mraa_gpio_dir_t) dir：GPIO 方向。

回傳：(mraa_result_t) 指令執行結果

mraa_result_t mraa_gpio_write (mraa_gpio_context dev, int value)

功能：設定 GPIO 針腳高電位或低電位。

參數：

a. (mraa_gpio_context) dev：代表 GPIO 針腳狀態內容。

b. (int) value : 欲寫入之電位高低 , 1 表示高電位、0 表示低電位。
回傳 : (mraa_result_t) 指令執行結果

int mraa_gpio_read (mraa_gpio_context dev)

功能 : 讀取 GPIO 針腳狀態為高電位還是低電位

參數 :

(mraa_gpio_context) dev : 指定讀取哪個 GPIO 針腳。

回傳 : (int) 電位高低。

針腳為高電位時回傳 1 , 當針腳為低電位時回傳0

擴充GPIO腳位數量

1. 遇到的問題

UpBoard所有的針腳只有 20 個 , 如果遇到欲讀取的bit超過 20 個的狀況 , 用一般的方法將無法全部進行輸入

2. 解決辦法

使用 Phillips 公司開發的 74HC165 晶片 來擴充輸入腳位的數量

74HC165晶片

1. 介紹

若是最直覺的方法 , 每當需要讀取一個 bit 的值 , 就必須使用一支針腳進行讀取 , 若要讀取的 bit 數很多 , 則針腳數量可能會不夠用 , 透過 74HC165 晶片可以將並行資料轉成串列資料 , 因此可以實現單一支針腳讀取 8 個 bit 的值

2. 原理

原先並行的 8 bit data 先傳入 74HC165 晶片 , 透過 74HC165 晶片中的暫存器與串列輸出 , 通過資料右移把暫存器內的值依序傳出



Step5: 為了待會產生正脈衝，將 CLK 調低

Step6: 讀取DATA的值

Step7: 讓 CLK 產生正脈衝，使 DATA 位移

重複Step5~7，直到所有暫存器的值讀完

Hands-on Practicing

1. 在終端中輸入 `ssh em_dev` 進入dev環境
2. 在終端中輸入 `cd Lab1-ENV_GPIO/Code` 進入Code資料夾
3. 依序輸入以下指令
 - (1) `chmod +x result_example/*`
 - (2) `scp result_example/* em_up:`
 - (3) `ssh em_up` (備註: 輸入此行後會進入dev_up環境)
4. 輸入 `ls` 查看檔案名稱
5. 輸入 `sudo ./<欲執行的程式名稱>`，以執行程式password:embedded (輸入密碼時終端不會顯示出來，輸入完後直接按enter即可)
6. 欲結束程式時按下 `ctrl+c` 退出程式，接著按下 `ctrl+d` 回到 `em_dev` 環境

Visual Studio Code

1. 開啟 Visual Studio Code (需安裝 遠端插件)
2. 點左下角的綠色按鍵
3. 在下拉選單中選：
 - a. "Remote SSH: Connect Current Window to Host..."
 - b. "em_dev"
 - c. 連線完成後左下角應如右圖：
4. 在上方選單開啟資料夾：
 - a. 點選 " File > Open Folder... "
 - b. 資料夾選 "Lab1-ENV_GPIO/Code/"
 - c. 按 OK
5. 按 `Ctrl-`` (Tab 上面那個鍵) 開啟內建終端
6. 在裡面執行 `sh ./set_env.sh` (一次性的初始化)
7. 修改程式後儲存 (內建檔案已寫好了程式骨架)
8. 依序輸入以下指令
 - a. `make bin/<程式名>`
 - b. `scp bin/* em_up:`
 - c. `ssh em_up` (備註: 輸入此行後會進入up環境)
9. 輸入`sudo ./<欲執行的程式名稱>`，以執行程式
10. 欲結束程式時按下 `ctrl+c` 退出程序，接著按下 `ctrl+d` 回到 `em_dev` 環境

作業繳交

在 DEV 上 cd 進入 Code/資料夾後
執行 `sh ./make_zip.sh`
這會產生 `archive.tar.gz`