1. One-Wire驅動:

* One-Wire的定義是主機和從機通過一根線進行通信，由DALLAS半導體公司推出的一種通信技術。優點在於單總線技術具有線路簡單、硬體開銷少、成本低。這裡將會學習如何在Linux下編寫One-Wire驅動。

1. Linux中One-Wire驅動實作:

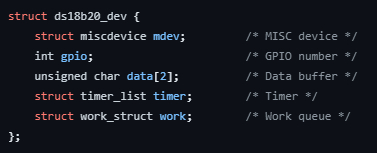
* One-Wire顧名思義就是只有一條線進行傳輸數據，也就是只有一條GPIO做輸入輸出。

1. 定義設備結構體，One-Wire被歸類在Linux系統中的MISC子系統。

(如下圖一，表一)

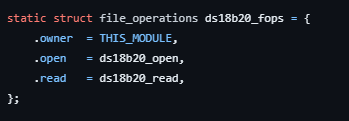
(表一: 設備結構體成員變數及其功能)

|  |  |  |
| --- | --- | --- |
| **成員** | **類型** | **描述** |
| mdev | struct miscdevice | MISC 設備 |
| gpio | int | GPIO 編號 |
| data | unsigned char[2] | 數據緩衝區 |
| timer | struct timer\_list | 計時器 |
| work | struct work\_struct | 工作隊列 |

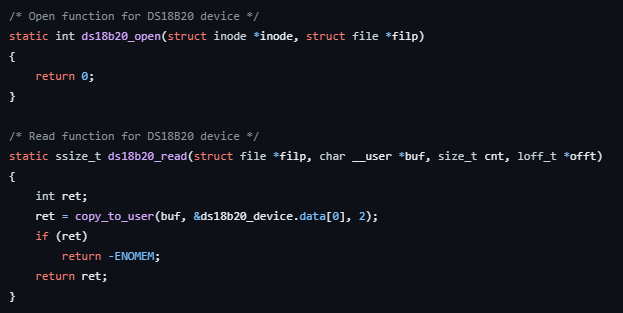


(圖一:設備結構體)

1. 接著定義file\_operations操作函式結構體，及其成員函數實現(如下圖二，圖三)



(圖二: file\_operations操作函式結構體內容)



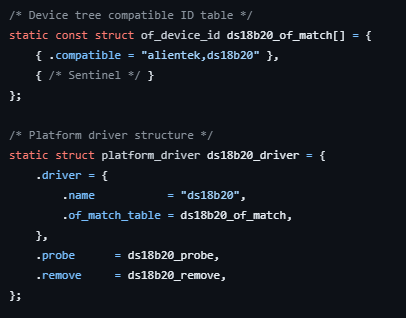
(圖三: file\_operations操作函式結構體成員函數實現)

1. 接著定義設備ID\_Table還有platform\_driver結構體。

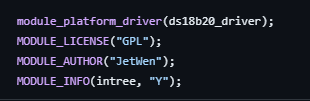
* ID\_Table用來匹配Device Tree中的設備。
* Platform\_driver則是定義匹配後的probe和remove函數。

(如下圖四)

* 透過module\_platform\_driver把ds18b20\_dirver加到Linux系統中。(如下圖五)



(圖四: ID\_Table和Platform\_driver內容定義)



(圖五:將ds18b20\_dirver加到Linux系統)

1. 實現Platform\_driver中的probe和remove函式(如下圖六)

* Probe函式內容實現(如下表二)
* Rmove函式內容實現(如下表二)

(表二Probe函式內容實現)

|  |  |
| --- | --- |
| **步驟** | **說明** |
| 1. 輸出信息 | 向Kernel日志輸出設備和驅動匹配成功的信息 |
| 2. 請求 GPIO | 請求與 DS18B20 傳感器相關的 GPIO 引腳，失敗則返回錯誤 |
| 3. 設置 miscdevice 結構 | 初始化 miscdevice 結構，設置設備名稱、次設備號和文件操作結構 |
| 4. 設置計時器 | 設置計時器，計時器在一秒鐘後觸發回調函數 |
| 5. 初始化工作結構 | 初始化工作結構，設置工作回調函數 |
| 6. 註冊 miscdevice | 註冊 miscdevice，使其在User Space中可用 |

(表三:Remove函式內容實現)

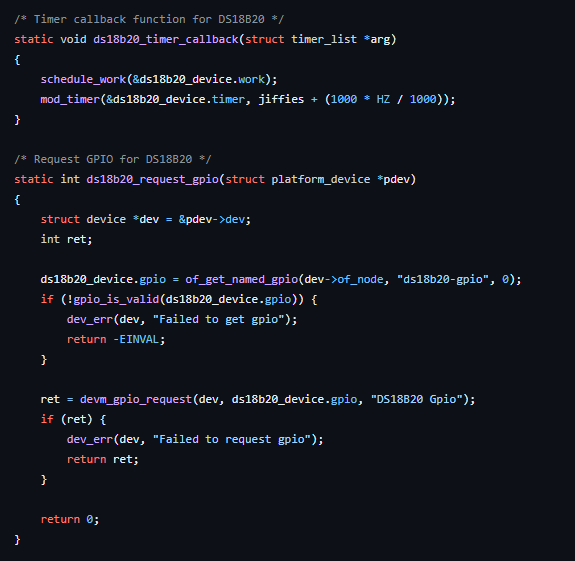
|  |  |
| --- | --- |
| **步驟** | **說明** |
| 1. 輸出信息 | 向內核日志輸出驅動程序已被移除的信息 |
| 2. 取消註冊 miscdevice | 取消註冊 miscdevice，使其在User Space中不可用 |
| 3. 刪除計時器 | 刪除計時器，停止計時器運行 |
| 4. 取消工作 | 取消工作隊列中的工作，確保沒有工作正在進行 |



(圖六:Remove和Probe函數實現)

1. 實現probe函式中使用到的函式

* ds18b20\_timer\_callback 函數: 在計時器觸發時被調用。它的主要作用是安排工作並重新設置計時器。(如下圖七，表四)
* ds18b20\_request\_gpio 函數: 用於請求與 DS18B20 傳感器相關的 GPIO 引腳。(如下圖七，表五)
* ds18b20\_work\_callback 函數: 在工作隊列中被調用，用於與 DS18B20 溫度傳感器進行通信，讀取數據。(如下圖八，表六)



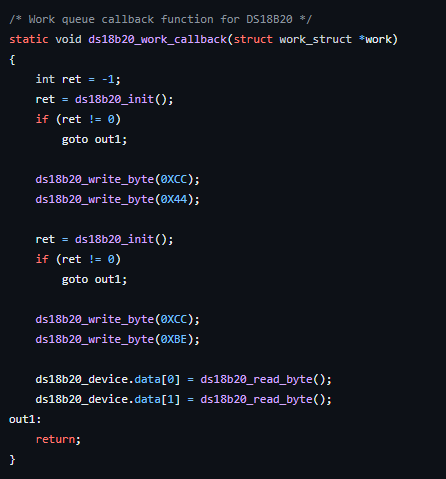
(圖七: ds18b20\_timer\_callback和ds18b20\_request\_gpio函數內容)

(表四: ds18b20\_timer\_callback 函數功能解析)

|  |  |
| --- | --- |
| **代碼** | **說明** |
| schedule\_work(&ds18b20\_device.work); | 將工作加入系統的工作隊列中 |
| mod\_timer(&ds18b20\_device.timer, jiffies + (1000 \* HZ / 1000)); | 設置計時器在一秒鐘後再次觸發 |

(表五: ds18b20\_request\_gpio 函數功能解析)

|  |  |
| --- | --- |
| **代碼** | **說明** |
| ds18b20\_device.gpio = of\_get\_named\_gpio(dev->of\_node, "ds18b20-gpio", 0); if (!gpio\_is\_valid(ds18b20\_device.gpio)) { dev\_err(dev, "Failed to get gpio"); return -EINVAL; } | 從設備樹中獲取 GPIO 編號，並檢查其有效性 |
| ret = devm\_gpio\_request(dev, ds18b20\_device.gpio, "DS18B20 Gpio"); if (ret) { dev\_err(dev, "Failed to request gpio"); return ret; } | 請求使用獲取到的 GPIO 引腳 |
| return 0; | 返回成功狀態 |

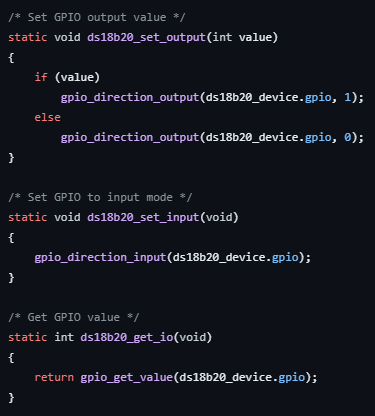


(圖八: ds18b20\_work\_callback 函數內容)

(表六: ds18b20\_work\_callback 函數功能解析)

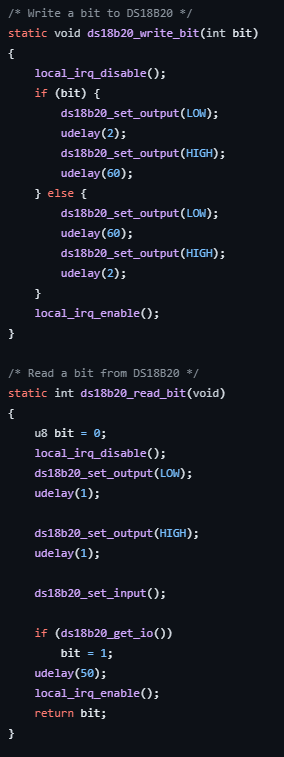
|  |  |
| --- | --- |
| **代碼** | **說明** |
| int ret = -1; ret = ds18b20\_init(); if (ret != 0) goto out1; | 初始化 DS18B20 傳感器，檢查是否成功，失敗則跳轉到 out1 |
| ds18b20\_write\_byte(0XCC); ds18b20\_write\_byte(0X44); | 發送命令，啟動溫度轉換 |
| ret = ds18b20\_init(); if (ret != 0) goto out1; | 再次初始化 DS18B20 傳感器，檢查是否成功，失敗則跳轉到 out1 |
| ds18b20\_write\_byte(0XCC); ds18b20\_write\_byte(0XBE); ds18b20\_device.data[0] = ds18b20\_read\_byte(); ds18b20\_device.data[1] = ds18b20\_read\_byte(); | 發送命令，讀取兩個字節的溫度數據，存儲在 ds18b20\_device.data 中 |
| out1: return; | 結束函數執行，無論成功還是失敗都跳轉到這裡並返回 |

1. 設定GPIO輸入輸出函式(如下圖九)

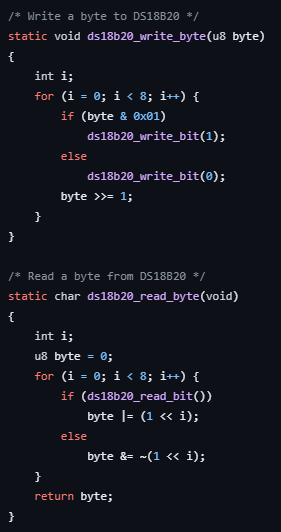


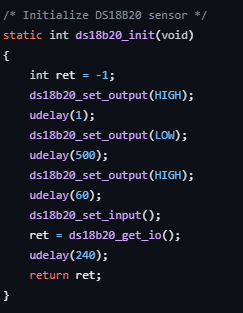
(圖九:GPIO輸入輸出設定函數)

1. 初始化DS18B20函數以及讀/寫數據函數(如下圖十)



(圖十:讀/寫單一數據函式)

  
(圖十一:讀/寫多個數據函式)



(圖十二:初始化DS18B20函式)

※總結:

以上就是One-Wire驅動開發整體流程，把握基本觀念就是GPIO，中斷，Timer，MISC子系統的應用。(如下表七)

(表七:One-Wire整體開發流程)

|  |  |
| --- | --- |
| **步驟** | **說明** |
| 1. 輸出信息 | 向內核日志輸出設備和驅動匹配成功的信息 |
| 2. 請求 GPIO | 請求與 DS18B20 傳感器相關的 GPIO 引腳，失敗則返回錯誤 |
| 3. 設置 miscdevice 結構 | 初始化 miscdevice 結構，設置設備名稱、次設備號和文件操作結構 |
| 4. 設置計時器 | 設置計時器，計時器在一秒鐘後觸發回調函數 |
| 5. 初始化工作結構 | 初始化工作結構，設置工作回調函數 |
| 6. 註冊 miscdevice | 註冊 miscdevice，使其在用戶空間中可用 |
| 7. 計時器回調 | 在計時器觸發時安排工作，並重新設置計時器 |
| 8. 獲取 GPIO | 從設備樹中獲取名為 "ds18b20-gpio" 的 GPIO 編號，並檢查其有效性 |
| 9. 請求 GPIO | 請求使用獲取到的 GPIO 引腳，檢查請求結果 |
| 10. 安排工作回調 | 初始化 DS18B20 傳感器，啟動溫度轉換，再次初始化 DS18B20 傳感器，讀取溫度數據 |
| 11. 移除驅動 | 取消註冊 miscdevice，刪除計時器，取消工作隊列中的工作 |