1. 專案31\_IIO:

* 隨著時代演進，ADC-DAC這類型傳感器越來越多也越來越多樣，而這接傳感器常常是鈄過SPI或I2C來進性數據的傳輸介面。Linux Kernel也為了方便管理這些日新月異的傳感器，提出IIO子系統。

1. IIO子系統基本介紹:

* IIO(Industrial I/O)，因為ADC-DAC這類傳感器大量應用在工業當中而得名，因此，IIO其實是專門設計給ADC-DAC這類傳感器的子系統。
* IIO子系統使用結構體iio\_dev來描述一個具體的IIO設備，此設備結構體定義在include/linux/iio/iio.h文件中。其中，iio\_dev支援幾個比較重要的成員變數。(如下表一整理)

(表一:成員變數及功能描述)

|  |  |
| --- | --- |
| **成員變數名稱** | **功能描述** |
| modes | 設備支持的模式 |
| currentmode | 當前模式 |
| buffer | 緩衝區 |
| buffer\_list | 當前匹配的緩衝區列表 |
| scan\_bytes | 捕獲到，並且提供給緩衝區的字節數 |
| available\_scan\_masks | 可選的掃描位掩碼，使用觸發緩衝區的時候可以通過設置掩碼來確定使能哪些通道，使能以後的通道會將捕獲到的數據發送到 IIO 緩衝區 |
| active\_scan\_mask | 緩衝區已經開啟的通道掩碼。只有這些使能了的通道數據才能被發送到緩衝區 |
| scan\_timestamp | 掃描時間戳，如果使能以後會將捕獲時間戳放到緩衝區裡面 |
| trig | IIO 設備當前觸發器，當使用緩衝模式的時候 |
| pollfunc | 一個函數，在接收到的觸發器上運行 |
| channels | IIO 設備通道，為 iio\_chan\_spec 結構體類型，稍後會詳細講解 IIO 通道 |
| num\_channels | IIO 設備的通道數 |
| name | IIO 設備名字 |
| info | iio\_info 結構體類型，這個結構體裡面有很多函數，需要驅動開發人員編寫，我們從User Space讀取。 |

* 其中modes表示設備支援的模式，所有支援模式整理如下表二。

(表二:成員變數modes支援的模式)

|  |  |
| --- | --- |
| **模式** | **功能描述** |
| INDIO\_DIRECT\_MODE | 提供 sysfs 接口。 |
| INDIO\_BUFFER\_TRIGGERED | 支持硬件緩衝觸發。 |
| INDIO\_BUFFER\_SOFTWARE | 支持軟件緩衝觸發。 |
| INDIO\_BUFFER\_HARDWARE | 支持硬件緩衝區。 |
| INDIO\_EVENT\_TRIGGERED | 支持事件觸發。 |
| INDIO\_HARDWARE\_TRIGGERED | 支持硬件觸發。 |
| INDIO\_ALL\_BUFFER\_MODES | 支持所有緩衝模式。 |
| INDIO\_ALL\_TRIGGERED\_MODES | 支持所有觸發模式。 |

1. IIO子系統常見API函式:

* 在使用之前要先申請iio\_dev，申請函數為iio\_device\_alloc。(如下表三)
* 完成以後就要初始化各種成員變數，初始化完成以後就需要將iio\_dev註冊到Kernel中，需要用到iio\_device\_register函數。(如下表三)
* 如果要註銷iio\_dev使用iio\_device\_unregister函數。(如下表三)
* iio\_info為結構體指針變數，這個是我們在編寫IIO驅動的時候需要著重去實現的，因為User Space對設備的具體操作最終都會反映到iio\_info裡面，iio\_info 結構體定義在 include/linux/iio/iio.h 中。(如下表三)

(表三:API函式及功能整理)

|  |  |
| --- | --- |
| **函式名稱** | **功能描述** |
| iio\_device\_alloc | 申請 iio\_dev，在使用之前需要先申請這個設備。 |
| iio\_device\_register | 將初始化完成的 iio\_dev 註冊到內核中。 |
| iio\_device\_unregister | 註銷 iio\_dev。 |
| iio\_info | 結構體指針變數，需要在編寫 IIO 驅動時著重實現，User Space對設備的具體操作最終都會反映到 iio\_info 裡面。 |

* 其中，iio\_info結構體中比較重要的成員函式如下表四整理。

(表四: iio\_info結構體成員函式及功能)

|  |  |
| --- | --- |
| **成員函數名稱** | **功能描述** |
| read\_raw | 讀取原始數據。 |
| write\_raw | 寫入原始數據。 |
| write\_raw\_get\_fmt | 獲取寫入原始數據的格式。 |

* iio\_chan\_spec結構體: IIO的核心就是通道，一個傳感器可能有多路數據，比如一個ADC芯片支持8路採集，那麼這個ADC就有8個通道。Linux Kernel使用iio\_chan\_spec結構體來描述通道，定義在 include/linux/iio/iio.h文件中。
* 而在iio\_chan\_spec結構體中有一個iio\_chan\_type結構體，列舉出了可以選擇的通道類型，定義在 include/uapi/linux/iio/types.h 文件裡面。

(如表五整理)

(表五:成員變數及其功能)

|  |  |
| --- | --- |
| **成員變數名稱** | **功能描述** |
| IIO\_VOLTAGE | 電壓通道 |
| IIO\_CURRENT | 電流通道 |
| IIO\_POWER | 功率通道 |
| IIO\_ACCEL | 加速度通道 |
| IIO\_ANGL\_VEL | 角速度通道 |
| IIO\_TEMP | 溫度通道 |
| IIO\_PRESSURE | 壓力通道 |
| IIO\_HUMIDITYRELATIVE | 相對濕度通道 |
| IIO\_LIGHT | 光強度通道 |
| IIO\_PROXIMITY | 接近度通道 |
| IIO\_MAGN | 磁場強度通道 |
| IIO\_INCLI | 傾斜角度通道 |
| IIO\_ROT | 旋轉角度通道 |
| IIO\_ACTIVITY | 活動狀態通道 |
| IIO\_STEPS | 步數通道 |
| IIO\_ENERGY | 能量通道 |
| IIO\_DISTANCE | 距離通道 |

(註:由此可以看出Linux線在支援許多種類的傳感器。)