Buffers

- struct iio_buffer â€” general buffer structure
- :c:func:`iio_validate_scan_mask_onehot` â€” Validates that exactly one channel is selected

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 6); *backlink*

  Unknown interpreted text role "c:func".

- :c:func:`iio_buffer_get` â€” Grab a reference to the buffer

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 8); *backlink*

  Unknown interpreted text role "c:func".

- :c:func:`iio_buffer_put` â€” Release the reference to the buffer

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 9); *backlink*

  Unknown interpreted text role "c:func".

The Industrial I/O core offers a way for continuous data capture based on a trigger source. Multiple data channels can be read at once from :file:`/dev/iio:device{X}` character device node, thus reducing the CPU load.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 11); *backlink*

Unknown interpreted text role "file".

IIO buffer sysfs interface

An IIO buffer has an associated attributes directory under :file:`/sys/bus/iio/iio:device{X}/buffer/*`. Here are some of the existing attributes:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 17); *backlink*

Unknown interpreted text role "file".

- :file:`length`, the total number of data samples (capacity) that can be stored by the buffer.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 21); *backlink*

  Unknown interpreted text role "file".

- :file:`enable`, activate buffer capture.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 23); *backlink*

  Unknown interpreted text role "file".

IIO buffer setup

The meta information associated with a channel reading placed in a buffer is called a scan element. The important bits configuring scan elements are exposed to userspace applications via the :file:`/sys/bus/iio/iio:device{X}/scan_elements/` directory. This directory contains attributes of the following form:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 28); ***backlink***

Unknown interpreted text role "file".

- :file:`enable`, used for enabling a channel. If and only if its attribute is non *zero*, then a triggered capture will contain data samples for this channel.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 34); ***backlink***

  Unknown interpreted text role "file".

- :file:`index`, the scan_index of the channel.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 37); ***backlink***

  Unknown interpreted text role "file".

- :file:`type`, description of the scan element data storage within the buffer and hence the form in which it is read from user space. Format is [be|le]:[s|u]bits/storagebits[Xrepeat][>>shift] .

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master) (Documentation) (driver-api) (iio)buffers.rst`, line 38); ***backlink***

  Unknown interpreted text role "file".

  - *be* or *le*, specifies big or little endian.
  - *s* or *u*, specifies if signed (2's complement) or unsigned.
  - *bits*, is the number of valid data bits.
  - *storagebits*, is the number of bits (after padding) that it occupies in the buffer.
  - *repeat*, specifies the number of bits/storagebits repetitions. When the repeat element is 0 or 1, then the repeat value is omitted.
  - *shift*, if specified, is the shift that needs to be applied prior to masking out unused bits.

For example, a driver for a 3-axis accelerometer with 12 bit resolution where data is stored in two 8-bits registers as follows:

```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|D3 |D2 |D1 |D0 | X | X | X | X | (LOW byte, address 0x06)
+---+---+---+---+---+---+---+---+

  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|D11|D10|D9 |D8 |D7 |D6 |D5 |D4 | (HIGH byte, address 0x07)
+---+---+---+---+---+---+---+---+
```

will have the following scan element type for each axis:

```
$ cat /sys/bus/iio/devices/iio:device0/scan_elements/in_accel_y_type
le:s12/16>>4
```

A user space application will interpret data samples read from the buffer as two byte little endian signed data, that needs a 4 bits right shift before masking out the 12 valid bits of data.

For implementing buffer support a driver should initialize the following fields in iio_chan_spec definition:

```
struct iio_chan_spec {
/* other members */
        int scan_index
        struct {
                char sign;
                u8 realbits;
                u8 storagebits;
                u8 shift;
                u8 repeat;
                enum iio_endian endianness;
               } scan_type;
       };
```

The driver implementing the accelerometer described above will have the following channel definition:

```
struct iio_chan_spec accel_channels[] = {
        {
                .type = IIO_ACCEL,
                .modified = 1,
                .channel2 = IIO_MOD_X,
                /* other stuff here */
                .scan_index = 0,
                .scan_type = {
                        .sign = 's',
                        .realbits = 12,
                        .storagebits = 16,
                        .shift = 4,
                        .endianness = IIO_LE,
                },
        }
        /* similar for Y (with channel2 = IIO_MOD_Y, scan_index = 1)
         * and Z (with channel2 = IIO_MOD_Z, scan_index = 2) axis
         */
 }
```

Here **scan_index** defines the order in which the enabled channels are placed inside the buffer. Channels with a lower **scan_index** will be placed before channels with a higher index. Each channel needs to have a unique **scan_index**.

Setting **scan_index** to -1 can be used to indicate that the specific channel does not support buffered capture. In this case no entries will be created for the channel in the scan_elements directory.

More details

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master)(Documentation)(driver-api)(iio)buffers.rst`, line 124)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: include/linux/iio/buffer.h
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\iio\(linux-master)(Documentation)(driver-api)(iio)buffers.rst`, line 125)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/iio/industrialio-buffer.c
   :export:
```