Data acquisition with the ADS1115 on the raspberry PI

Generated by Doxygen 1.8.17

Chapter 1

rpi_ads1115

Raspberry PI C++ library for the ADS1115

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 ADS1115rpi Class Reference

This class reads data from the ADS1115 in the background (separate thread) and calls a callback function whenever data is available.

`#include <ads1115rpi.h>`

Public Member Functions

- ADS1115rpi ()
  Constructor with the spiDevice.
- ~ADS1115rpi ()
  Destructor which makes sure the data acquisition has stopped.
- virtual void hasSample (float sample)=0
  Called when a new sample is available.
- void setChannel (ADS1115settings::Input channel)
  Selects a different channel at the multiplexer while running.
- void start (ADS1115settings settings=ADS1115settings())
  Starts the data acquisition in the background and the callback is called with new samples.
- void stop ()
  Stops the data acquistion.

3.1.1 Detailed Description

This class reads data from the ADS1115 in the background (separate thread) and calls a callback function whenever data is available.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 ADS1115rpi()

`ADS1115rpi::ADS1115rpi ( )`

Constructor with the spiDevice.

The default device is /dev/spidev0.0.

Parameters

spiDevice	The raw /dev spi device.

3.1.3 Member Function Documentation

3.1.3.1 hasSample()

```
virtual void ADS1115rpi::hasSample (
            float sample )  [pure virtual]
```

Called when a new sample is available.

This needs to be implemented in a derived class by the client. Defined as abstract.

3.1.3.2 setChannel()

```
void ADS1115rpi::setChannel (
            ADS1115settings::Input channel )
```

Selects a different channel at the multiplexer while running.

Call this in the callback handler hasSample() to cycle through different channels.

3.1.3.3 start()

```
void ADS1115rpi::start (
            ADS1115settings settings = ADS1115settings() )
```

Starts the data acquisition in the background and the callback is called with new samples.

Parameters

samplingRate	The sampling rate of the ADC.

The documentation for this class was generated from the following file:

- ads1115rpi.h

3.2 ADS1115settings Struct Reference

ADS1115 initial settings when starting the device.

```
#include <ads1115rpi.h>
```

Public Types

- enum SamplingRates {
  FS8HZ = 0, **FS16HZ** = 1, **FS32HZ** = 2, **FS64HZ** = 3,
  FS128HZ = 4, **FS250HZ** = 5, **FS475HZ** = 6, **FS860HZ** = 7 }

  Sampling rates.
- enum PGA { **FSR2_048** = 2, **FSR1_024** = 3, **FSR0_512** = 4, **FSR0_256** = 5 }

  Gains of the PGA.
- enum Input { **AIN0** = 0, **AIN1** = 1, **AIN2** = 2, **AIN3** = 3 }

  Channel indices.

Public Attributes

- int i2c_bus = 1

  I2C bus used (99% always set to one)
- uint8_t address = DEFAULT_ADS1115_ADDRESS

  I2C address of the ads1115.
- SamplingRates samplingRate = FS8HZ

  Sampling rate requested.
- PGA pgaGain = FSR2_048

  Requested gain.
- Input channel = AIN0

  Requested input channel (0 or 1)
- bool initPIGPIO = true

  If set to true the pigpio is initialised.
- int drdy_gpio = DEFAULT_DATA_READY_GPIO

  Default GPIO pin for data ready.

3.2.1 Detailed Description

ADS1115 initial settings when starting the device.

The documentation for this struct was generated from the following file:

- ads1115rpi.h

Index