Utilities

Table of contents

FAPI and Matlab interface utilities

Data storing utilities

CUDA utilities

FAPI and Matlab interface utilities

The FAPI module contains various utilities for handling the interface between the PUSCH database schema (SCF FAPI) and cuPHY.

aerial.util.fapi.dmrs_fapi_to_bit_array(*dmrs_symb_pos*)

Convert the DMRS symbol position decimal value to a bit array.

Parameters

dmrs_symb_pos (*np.uint16*) – DMRS symbol position decimal value as defined in SCF FAPI.

Returns

A bit array to be used for cuPHY interface, indicating the positions of DMRS symbols. The first bit corresponds to OFDM symbol 0.

Return type

list

aerial.util.fapi.dmrs_bit_array_to_fapi(*x*)

Convert a bit array to DMRS symbol position decimal value.

Parameters

x (*list*) – A bit array to be used for cuPHY interface, indicating the positions of DMRS symbols. The first bit corresponds to OFDM symbol 0.

Returns

DMRS symbol position decimal value as defined in SCF FAPI.

Return type

np.uint16

aerial.util.fapi.dmrs_fapi_to_sym(*dmrs_symb_pos*)

Convert the DMRS symbol position decimal value to a list of DMRS symbol indices.

Parameters

dmrs_symb_pos (*np.uint16*) – DMRS symbol position decimal value as defined in SCF FAPI.

Returns

A list of DMRS symbol indices.

Return type

list

aerial.util.fapi.mac_pdu_to_bit_array(*mac_pdu*)

Convert MAC PDU bytes to a bit array.

Parameters

mac_pdu (*list*) – A list of bytes, the content of the MAC PDU.

Returns

The same MAC PDU as a bit array, i.e. the bytes are converted to a list of bits.

Return type

list

aerial.util.fapi.bit_array_to_mac_pdu(*bits*)

Convert a bit array to MAC PDU bytes.

Parameters

bits (*list*) – A MAC PDU as a bit array.

Returns

A list of bytes corresponding to the above MAC PDU.

Return type

list

Data storing utilities

class aerial.util.data.PuschRecord

Implements column schema of a PUSCH dataframe row.

The *PuschRecord* includes fields collected from the data collection agent, and SCF FAPI message content for the PUSCH channels from UL_TTI.request, RxData.indication, and CRC.indication.

Parameters

- **SFN** – System Frame Number. Value: 0 - 1023.
- **Slot** – Slot number. Value: 0 - 159.
- **nPDUs** – Number of PDUs that were included in the UL_TTI.request message.
- **RachPresent** –

  Indicates if a RACH PDU was included in the UL_TTI.request message.

  - 0: No RACH in this slot.
  - 1: RACH in this slot.
- **nULSCH** – Number of ULSCH PDUs that were included in the UL_TTI.request message. Value: 0 - 255.
- **nULCCH** – Number of ULCCH PDUs that were included in the UL_TTI.request message. Value: 0 - 255.
- **nGroup** – Number of UE Groups that were included in the UL_TTI.request message. Value: 0 - 8.
- **PDUSize** – Size of the PDU control information (in bytes). This length value includes the 4 bytes required for the PDU type and PDU size parameters. Value: 0 - 65535.

- **nUE** – Number of UEs in this group. For SU-MIMO, one group includes one UE only. For MU-MIMO, one group includes up to 12 UEs. Value: 1 - 6, None if nGroup = 0.

- **pduIdx** – This value is an index for number of PDU identified by nPDU in the UL_TTI.request message. Value: 0 - 255, None if nGroup = 0.

- **pduBitmap** –

  Bitmap indicating presence of optional PDUs.

  - Bit 0: puschData (Indicates data is expected on the PUSCH).

  - Bit 1: puschUci (Indicates UCI is expected on the PUSCH).

  - Bit 2: puschPtrs (Indicates PTRS included (FR2)).

  - Bit 3: dftsOfdm (Indicates DFT S-OFDM transmission).

  - All other bits reserved.

- **RNTI** – The RNTI used for identifying the UE when receiving the PDU. Value: 1 - 65535.

- **Handle** – An opaque handling returned in the RxData.indication and/or UCI.indication message.

- **BWPSize** – Bandwidth part size [TS38.213 sec12]. Number of contiguous PRBs allocated to the BWP. Value: 1 - 275.

- **BWPStart** – Bandwidth part start RB index from reference CRB [TS38.213 sec 12]. Value: 0 - 274.

- **SubcarrierSpacing** – SubcarrierSpacing [TS38.211 sec 4.2]. Value: 0 - 4.

- **CyclicPrefix** –

  Cyclic prefix type [TS38.211 sec 4.2].

  - 0: Normal

  - 1: Extended

- **targetCodeRate** – Target coding rate [TS38.214 sec 6.1.4.1]. This is the number of information bits per 1024 coded bits expressed in 0.1 bit units.
- **qamModOrder** –

  QAM modulation [TS38.214 sec 6.1.4.1]. Values:

  - 2,4,6,8 if transform precoding is disabled.
  - 1,2,4,6,8 if transform precoding is enabled.
- **mcsIndex** – MCS index [TS38.214, sec 6.1.4.1], should match value sent in DCI. Value: 0 - 31.
- **mcsTable** –

  MCS-Table-PUSCH [TS38.214, sec 6.1.4.1]. Value:

  - 0: notqam256 [TS38.214, table 5.1.3.1-1].
  - 1: qam256 [TS38.214, table 5.1.3.1-2].
  - 2: qam64LowSE [TS38.214, table 5.1.3.1-3].
  - 3: notqam256-withTransformPrecoding [TS38.214, table 6.1.4.1-1].
  - 4: qam64LowSE-withTransformPrecoding [TS38.214, table 6.1.4.1-2].
- **TransformPrecoding** –

  Indicates if transform precoding is enabled or disabled [TS38.214, sec 6.1.4.1] [TS38.211 6.3.1.4].

  - 0: Enabled
  - 1: Disabled
- **dataScramblingId** – dataScramblingIdentityPdsch [TS38.211, sec 6.3.1.1]. It equals the higher-layer parameter Data-scrambling-Identity if configured and the RNTI equals the C-RNTI, otherwise L2 needs to set it to physical cell ID. Value: 0 - 65535.

- **nrOfLayers** – Number of layers [TS38.211, sec 6.3.1.3]. Value: 1 - 4.
- **ulDmrsSymbPos** – DMRS symbol positions [TS38.211, sec 6.4.1.1.3 and Tables 6.4.1.1.3-3 and 6.4.1.1.3-4]. Bitmap occupying the 14 LSBs with bit 0 corresponding to the first symbol and for each bit, value 0 indicates no DMRS and value 1 indicates DMRS.
- **dmrsConfigType** –

  UL DMRS config type [TS38.211, sec 6.4.1.1.3].

  - 0: type 1
  - 1: type 2
- **ulDmrsScramblingId** – UL-DMRS-Scrambling-ID [TS38.211, sec 6.4.1.1.1]. If provided and the PUSCH is not a msg3 PUSCH, otherwise, L2 should set this to physical cell ID. Value: 0 - 65535.
- **puschIdentity** – PUSCH-ID [TS38.211, sec 6.4.1.1.2]. If provided and the PUSCH is not a msg3 PUSCH, otherwise, L2 should set this to physical cell ID. Value: 0 - 1007.
- **SCID** – DMRS sequence initialization [TS38.211, sec 6.4.1.1.1]. Should match what is sent in DCI 0_1, otherwise set to 0. Value : 0 - 1.
- **numDmrsCdmGrpsNoData** – Number of DM-RS CDM groups without data [TS38.212 sec 7.3.1.1]. Value: 1 - 3.
- **dmrsPorts** –

  DMRS ports. [TS38.212 7.3.1.1.2] provides description between DCI 0-1 content and DMRS ports. Bitmap occupying the 11 LSBs with bit 0 corresponding to antenna port 1000 and bit 11 corresponding to antenna port 1011 and for each bit:

  - 0: DMRS port not used.
  - 1: DMRS port used.
- **resourceAlloc** –

Resource Allocation Type [TS38.214, sec 6.1.2.2].

  - 0: Type 0.

  - 1: Type 1.

- **rbBitmap** – For resource allocation type 0. [TS38.214, sec 6.1.2.2.1] [TS 38.212, 7.3.1.1.2] bitmap of RBs, 273 rounded up to multiple of 32. This bitmap is in units of VRBs. LSB of byte 0 of the bitmap represents the first RB of the BWP. Each element is of type *numpy.uint8*.

- **rbStart** – For resource allocation type 1. [TS38.214, sec 6.1.2.2.2]. The starting resource block within the BWP for this PUSCH. Value: 0 - 274.

- **rbSize** – For resource allocation type 1. [TS38.214, sec 6.1.2.2.2]. The number of resource block within for this PUSCH. Value: 1 - 275.

- **VRBtoPRBMapping** –

  VRB to PRB mapping [TS38.211, sec 6.3.1.7].

  - 0: Non-interleaved.

  - 1: Interleaved.

- **FrequencyHopping** –

  For resource allocation type 1, indicates if frequency hopping is enabled. [TS38.212, sec 7.3.1.1] [TS38.214, sec 6.3].

  - 0: Disabled.

  - 1: Enabled.

- **txDirectCurrentLocation** – The uplink Tx Direct Current location for the carrier. Only values in the value range of this field between 0 and 3299, which indicate the subcarrier index within the carrier corresponding to the numerology of the corresponding uplink BWP and value 3300, which indicates “Outside the carrier” and value 3301, which indicates “Undetermined position within the carrier” are used. [TS38.331, UplinkTxDirectCurrentBWP IE]. Value: 0 - 4095.

- **uplinkFrequencyShift7p5khz** –

  Indicates whether there is 7.5 kHz shift or not. [TS38.331, UplinkTxDirectCurrentBWP IE].

  - 0: False.
  - 1: True.

- **StartSymbolIndex** – Start symbol index of PUSCH mapping from the start of the slot, S. [TS38.214, Table 6.1.2.1-1]. Value: 0 - 13.

- **NrOfSymbols** – PUSCH duration in symbols, L. [TS38.214, Table 6.1.2.1-1]. Value: 1 - 14.

- **puschData** – See SCF FAPI 10.02, Table 3-47. dict{‘cbPresentAndPosition’: array([], dtype=int32), ‘harqProcessID’: np.uint8, ‘newDataIndicator’: np.uint8, ‘numCb’: np.uint8, ‘rvIndex’: np.uint8, ‘TBSize’: np.uint32}

- **puschUci** – See SCF FAPI 10.02, Table 3-48.

- **puschPtrs** – See SCF FAPI 10.02, Table 3-49.

- **dftsOfdm** – See SCF FAPI 10.02, Table 3-50.

- **Beamforming** – See SCF FAPI 10.02, Table 3-53.

- **HarqID** – HARQ process ID. Value: 0 - 15.

- **PDULen** – Length of PDU in bytes. A length of 0 indicates a CRC or decoding error.

- **UL_CQI** – SNR.

- **TimingAdvance** – Timing advance.

- **RSSI** – RSSI. See SCF FAPI 10.02 Table 3-16 for RSSI definition.

- **macPdu** – Contents of MAC PDU. Each element is of type *numpy.uint8*.

- **TbCrcStatus** –

Indicates CRC result on TB data. Each element is of type *numpy.uint8*.

  - 0: Pass.
  - 1: Fail.

- **NumCb** – If CBG is not used this parameter can be set to zero. Otherwise the number of CBs in the TB. Value: 0 - 65535.
- **CbCrcStatus** –

  Byte-packed array where each bit indicates CRC result on CB data. Each element is of type *numpy.uint8*.

  - 0: Pass.
  - 1: Fail.
  - None if NumCb = 0.

- **rx_iq_data_filename** – Filename of the received OFDM IQ data file. This file contains the complex OFDM slot data as a frequency x time x antenna numpy array.
- **user_data_filename** – Filename of the user data file. This file may contain for example ground truth data.
- **errInd** – Freeform error indication message.

Notes

The PDULen field is 32 bits whereas SCF FAPI 10.02 incorrectly uses 16 bits. Using 32 bits allows MAC PDUs larger than 65535 bytes.

static from_series(*series*)

Create a PuschRecord from a Pandas Series entry (e.g. a DataFrame row).

Parameters

series (*pandas.Series*) – The input dataframe row.

Returns

The PUSCH record built from the given Pandas Series.

Return type

PuschRecord

static columns()

Return the field names of PuschRecord.

Return type

Tuple

aerial.util.data.save_pickle(*data*, *filename*, *s3=None*)

Save the data in a pickle file either locally or on S3.

Parameters

- **data** (*np.ndarray or dict*) – The data to be saved.
- **filename** (*str*) – Full path of the file to be used.
- **s3** (*s3fs.S3FileSystem*) – The S3 filesystem to be used. Set to `None` for local filesystem.

Return type

None

aerial.util.data.load_pickle(*filename*, *s3=None*)

Load data from a pickle file, either a local file or on S3.

Parameters

- **filename** (*str*) – Full path of the file to be used.
- **s3** (*s3fs.S3FileSystem*) – The S3 filesystem to be used. Set to `None` for local filesystem.

Returns

The loaded data.

Return type

np.ndarray or dict

CUDA utilities

aerial.util.cuda.get_cuda_stream()

Return a CUDA stream.

Returns

A new CUDA stream.

Return type

cudart.cudaStream_t

aerial.util.cuda.check_cuda_errors(*result*)

Check CUDA errors.

Parameters

result (*cudart.cudaError_t*) – CUDA error value.

Return type

Any