Documentation

1/4/2021

s.dong@mails.ccnu.edu.cn

Central China Normal University

Documentation

1. Readme.rst and Paper(soon).
2. Document in code

    Software**: reStructuredText** (reST) and **Sphinx**.

    - Finished
    - Viewable both on Internet (some bug not cleared) and local.
    - https://jadepix3-software.readthedocs.io/en/latest/README.html

    Firmware**: Doxygen**.

    - Not Started yet

JadePix3 Software

docs passing

This is the software for Jadepix3 verification. The software is designed based on IPbus uHAL. The functions implemented:

- Configure the DAC70004 on the FMC(HPC) daughter PCB.
- Configure the JadePix3 via SPI.
- Configure each pixel in the JadePix3.
- Rolling shutter setup.
- Global shutter setup.
- Data acquisition.
- Data analysis by the ROOT.

Install libraries and requirements

Note

Please use **Python3**

```
pip3 install -r requirements.txt
pip3 install -e .
```

```python
def cmd(self, wr, cmd, chn, din, mode):
    """
    DAC commands, 32 bits.

    D31-D29: Don't cares.

    :param wr: D28: R/W. 0 = Write, 1 = Read.
    :param cmd: D27-D24: Command Bits
    :param chn: D23-D20: Channel Address Bits
    :param din: D19-D04: 16/14/12-Bit DAC Data left alligned/Power Down Bits/Device Ready bit
    :param mode: D03-D00: Mode Bits
    :return:
    """
    if cmd not in [DAC70004_CMD_WR_BUF, DAC70004_CMD_UPDATE_CHN, DAC70004_CMD_W_UPDATE_ALL, DAC70004_CMD_W_UPDATE,
                   DAC70004_CMD_POWER_ONOFF, DAC70004_CMD_CLR_MODE, DAC70004_CMD_LDAC_REG, DAC70004_CMD_SOFT_RST,
                   DAC70004_CMD_DIS_SDO, DAC70004_CMD_RESERVED_1, DAC70004_CMD_SCL_REG, DAC70004_CMD_SOFT_CLR,
                   DAC70004_CMD_RESERVED_2, DAC70004_CMD_STATUS_REG, DAC70004_CMD_NOP, DAC70004_CMD_RESERVED_3]:
        raise ValueError('Unexpected cmd value {0}.'.format(cmd))
    send_data = (wr << 28) + (cmd << 24) + (chn << 20) + (din << 4) + mode
    return self.write_data(send_data)
```

reStrcturedText in code and the html display.

class **lib.dac70004_device.Dac70004Device**(*ipbus_link*) [source]

Bases: `object`

static **anaVal_2_digVal**(*anaVal*) [source]

Static method, calculate the digital number for giving analog voltage.

Parameters: **anaVal** – Maximum equals to reference voltage: 2.5V

Returns:

cmd(*wr*, *cmd*, *chn*, *din*, *mode*) [source]

DAC commands, 32 bits.

D31-D29: Don't cares.

Parameters:

- **wr** – D28: R/W. 0 = Write, 1 = Read.
- **cmd** – D27-D24: Command Bits
- **chn** – D23-D20: Channel Address Bits
- **din** – D19-D04: 16/14/12-Bit DAC Data left alligned/Power Down Bits/Device Ready bit
- **mode** – D03-D00: Mode Bits

Returns:

is_busy() [source]