1 APDPI

The Avalanche Photodiode Power Interface APDPI is a standart NIM module that supplies power for the SiPMs, amplifier circuits and logic of the frontend electronics. Figure [??] shows the front panel of an APDPI module: On the top there is a DSUB-9 connector consisting of a VBIAS, V+, V-, GND, RTX+ and RTX- output. Below the connector there are two switches BIAS and POWER and again below that a USB B plug.

1. The $+12\,V$ line of the NIM backend is fed to a DC-DC converter that is adjustable via a potentiometer located next to the converter. The converter takes $10.8\,V..13.2\,V$ and outputs $0\,V..180\,V@15\,mA$ to the VBIAS line. The switch BIAS is located directly below the DSUB-9 connector and switches VBIAS on and off. The typical VBIAS voltage currently used is $82\,V$.

2. V+ and V- supply a $\pm3.3\,V@1A$ power source respectively for the amplifier and logic circuits of the frontend electronics. There are two voltage regulators which use the $\pm6\,V$ lines of the NIM backend to generate the $\pm3.3\,V$ outputs. Below the BIAS switch there is also a POWER switch located for V+ and V-.

3. GND is the reference potential for VBIAS as well as for V+ and V-. Is uses the ground line of the NIM backend and is also connected to the casing to reduce noise.

4. RTX+ and RTX- are the differential EIA-485 compliant outputs used for communication with and slow control of the frontend electronics. The EIA-485 or RS-485 defines a physical layer that uses a differential signal for communcation thus providing a stable signal over distances of up to $1.2\,km$. The data link layer is compliant to microcontroller friendly USART which simplifies the whole commucation chain in hardware. For more details on RS-485 see [??]. A computer can be connected to the APDPI via a USB B plug located beneath the POWER switch. The USB signal is mapped to USART by a *FT232RL* chip [??] and again converted to RS-485 by a *MAX3086E* chip [??] thus the slow control can be accessed with a simple FTDI serial driver [??].

1.1 SiPM Bus

The SiPM Bus defines the application layer for the serial communication with the frontend electronics. The bus is host driven, so the bus master has to query for data from the slaves. To ensure the correct understanding of the sent commands each individual byte is echoed by the slave and should be checked. To start the communication the bus master sends the start delimiter character '<' (ASCII 0x3C) and claims the bus (start delimiter will not be echoed). Any further try to claim the bus will fail because the bus is now busy. The start delimiter then is followed by a unique 8 bit slave address which is echoed by the slave. The slave is now addressed and is ready to receive commands. For a complete list of implemented commands and a short description see table [**??**]. Every command and parameter needed by the command will be echoed bytewise aswell. In addition to the echos the slave will send the End Of Message (EOM) string "\n \r *" (ASCII 0x0A, 0x0D, 0x2A) after receiving a complete command line including all parameters.

Currently the transmission rate is limited to 9600 BAUD to reduce power consumption of the microcontroller on the frontend electronics and the framing is set to 8N1 meaning eight bits of data per frame, no parity bit, one stop bit to maximize data throughput.

2 MPPC Dou Controller

2.1 Temperature Measurement

2.2 Bias Voltage Supply

2.3 Amplifier Circuits

2.4 Overview Of Technical Characteristics

3 Appendix: Circuit Diagrams

3.1 APDPI

3.2 MPPC Duo Controller