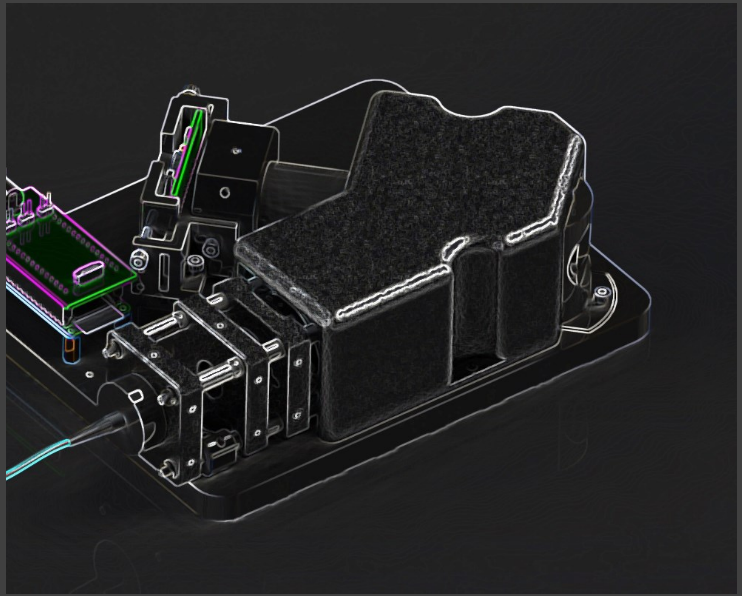
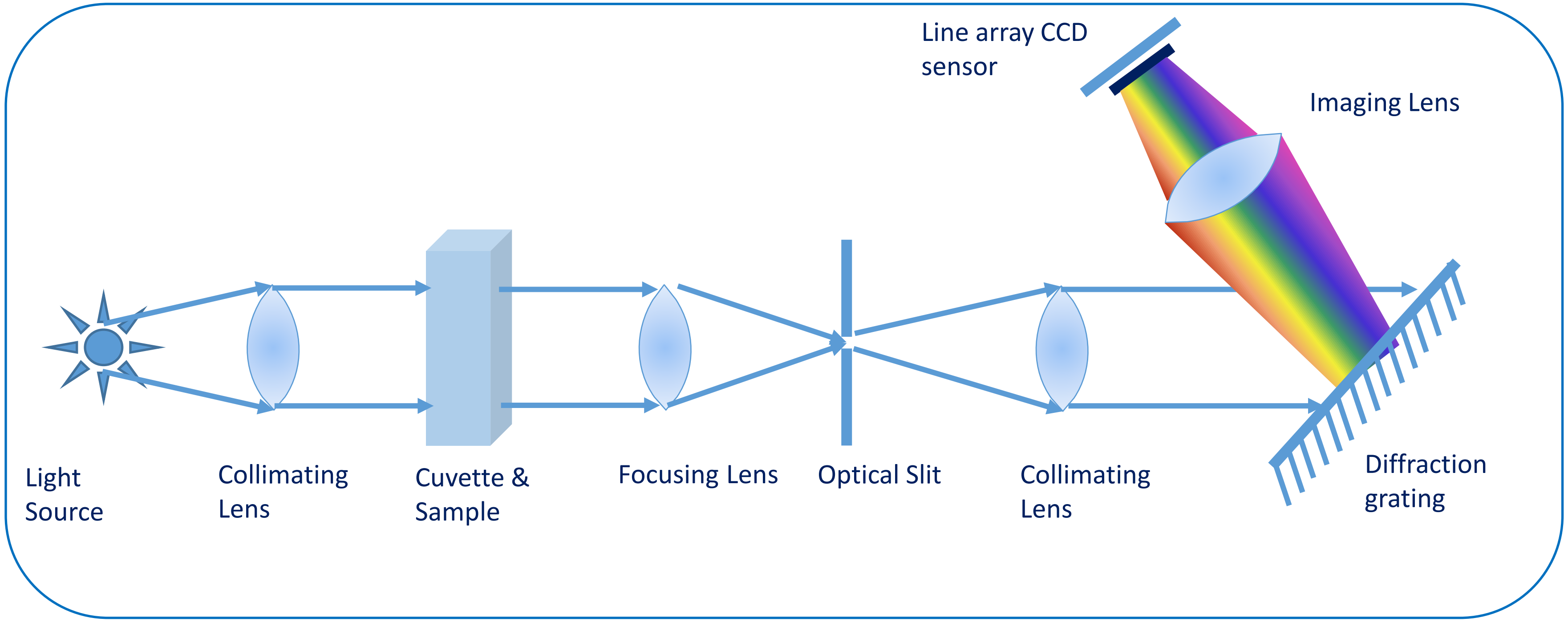


Creato Spectrometer

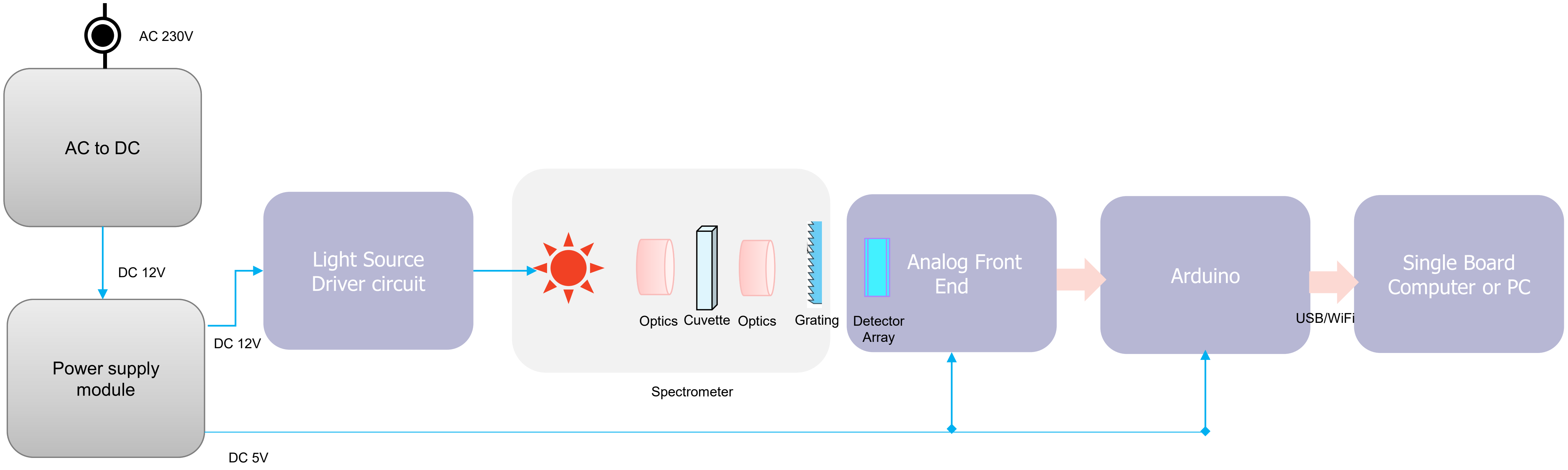
Module Characterisation

10-April-2024

VIS-NIR Grating Spectrometer: Block Diagram



System Block Diagram



Data Flow and throughput bottlenecks

- At 2 MHz Sensor clock, and 1 MHz data rate and with say 2 bytes per data, the data rate required at the source side will be 2MBytes/Sec
- We will have to clock down substantially for the Arduino to handle a data which is feasible to digitize

- MKR1000 features a 12 bit, 350ksps multi channel ADC
- Will have to characterise the noise floor of the detector to see if 12 bit is enough
- Achieving 350ksps with triple buffering will be a challenge and will be good to measure practical limits
- 32K SRAM is the limit with respect to memory

- MKR1000 does not have native USB and we are using a serial port over USB
- Arduino IDE and the serial port monitor has limitations with respect to baud rate
- We need to pump data from Arduino at the maximum possible rate and validate the data is reaching host side with out any frame errors

Clock Characteristics (Ta = 25°C) (3.0 V ≤ VAD = VDD ≤ 4.0 V)

For best performance, the device should be used within the Recommended Operating Conditions.

TCD1103 clock characteristics

Characteristics	Symbol	Min	Typ.	Max	Unit
Master clock pulse frequency	$f_{\phi M}$	0.4	2.0	4.0	MHz
Data rate	f_{DATA}	0.2	1.0	2.0	MHz

Test Stub: 1

Arduino to PC data transfer

a) Setup Arduino to fastest possible data rate
b) Create a known pattern (Sin look up table or similar predicable data)
c) Transfer data from Arduino to PC over serial port for a specific duration
d) Store the data in a file in PC
e) Write a script to validate the data
f) Automate the test rig and validate the performance and characterise the maximum throughput achievable

Test Stub: 2

Serial Port: DMA transfer

a) In test stub 1, it is possible to use the MCU to dedicate the CPU cycles only for data transfer
b) However, CPU will eventually have to be freed up for other tasks like buffer management, interrupt management etc
c) We will have to use a DMA to transfer data to serial port and also implement a triple buffering's to automate data movement from Arduino to PC
d) There is an event manager in SAMD21, which needs to be used to trigger the DMA transfer
e) The test stub should facilitate data transfer to PC using DMA in Arduino
f) Once the measurement is completed, we can characterise the data throughput realised using Arduino sending data employing DMA

Test Stub: 3

ADC throughput

a) Theoretical data throughput of the ADC is 350 ksps, but it will be difficult to achieve the same in practical conditions
b) We have to ensure that the digital conversion is error free and is synchronised with the sensor clocks
c) Using a DAC or a signal generator, we can create known signals and using the ADC, we can capture and send to PC for further analysis
d) A test stub featuring the ADC throughput validation will ensure that the ADC is working in synchronisation with the sensor clocks

System Test Stub

Simulating Spectrometer on Arduino

a) It will take time to complete the embedded side coding to clock the sensor, capture data, pre-process and send it to PC side

b) We will have to create a test stub, mimicking the actual spectrometer

c) Setup side

   i. Spectral averages

   ii. Integration time

   iii. Box car filtering

   iv. Single/Multiple spectra's

d) Spectral Acquisition

   i. Pre-view: Acquire and plot the spectra as per the settings in section c

   ii. Acquire: Record the spectra and save it to file on PC side

e) To achieve this, we need to have a draft protocol designed and also the format for spectral storage needs to be decided

Action Item

- Design document for System Test Stub: Application team
- Design document for the data throughput : Embedded team