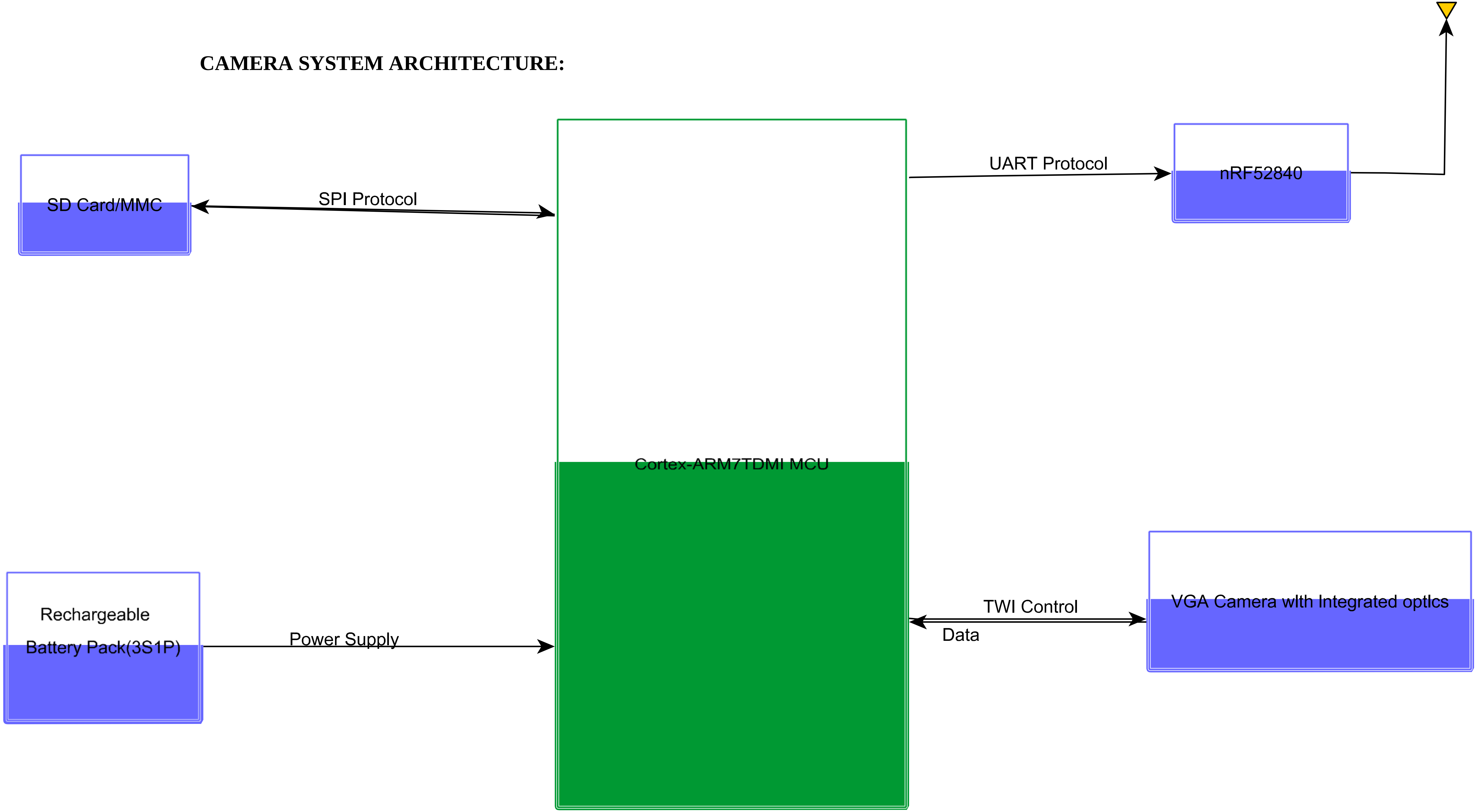
CAMERA SYSTEM ARCHITECTURE:



Hardware Selection:

Microcontroller: MCU for this architecture is preferably STM32 can be used because it is HAL Layer drivers and easy compared to other microcontrollers and support for STM series microcontrollers is high in the open source compared to other microcontrollers, different peripherals can be integrated at once and cost wise cheaper or else any ARM7 core microcontroller.

Battery Pack: Battery pack of 3S1P configuration with BMS can power up the all the modules in the above architecture approx. for 12hrs, this battery supports fast charging within 40-90min and also available at cheaper prices. So, I choose this battery pack for this architecture.

SD Card/MMC Card: SD Card/MMC Card is best as per my choice because it supports FATFS file system and also easily interfaced with the STM series of MCU's and also it can store large amount of data whenever authorized person is unavailable and now a days it is cheaper even cost wise.

Lens:

Fixed Focal Lens: As the camera needs to cover the fixed space inside the retail shop fixed local lens are best suitable as per my view.

Wide Angle Lens: Wide angle lens can be best preferred because they can cover larger area of the shop with less number of security cams. Considering other features of camera Aperture of camera must be small to cover large area clearly whereas wide aperture makes the object clear but background blurry.

nRF52840(BLE): Unlike classic Bluetooth, BLE consumes less power and can stay in sleep mode unless the connection is initiated, it is much easier to segregate the data using BLE. It is more widely used technology in industries. So, due to this reason I choose to go with BLE.

Communication Protocols:

UART Protocol: The protocol used for communication between BLE and MCU is UART Protocol.
UART stands for **Universal Asynchronous Receiver Transmitter**.

- UART is serial communication of digital data.
- It is a two wire
- In this both data format and speed are configurable.
- No clock signal is required for UART to communicate or synchronize between Rx and Tx.

Data Synchronization:

- Data synchronization in UART is achieved with same BAUD rate between Tx and Rx.
  BAUD Rate: It means rate of transmission speed which is measured in bits per second(bps).
  Which can be from 2400, 4800, 9600, 19200, 38400, 115200… bps.
- Data is also synchronized using start and stop bits in the frame.

Data Format in UART:

Start bit + data bits + parity bits + stop bits

Start bit: Start bit is a logic low. The bus is pulled low by the Tx as the communication begins. This indicates the start of communication.
Data bits: The data bits in UART are configurable it can vary from 5/6/7/8/9 data bits can be transmitted in a single frame.
Parity bits: The parity bit can be EVEN/ODD/NO Parity
Even Parity - 0
Odd Parity – 1
Stop bits: The stop bits can be 1/1.5/2 stop bits depending upon the user. Basically, this indicates the stop of communication.

UART Configuration: UART can be configured as SIMPLEX (or) HALF DUPLEX (or) FULL DUPLEX

SIMPLEX: This means communication is only one way.
HALF DUPLEX: This means communication is two way but not at the same time.
FULL DUPLEX: This means communication is two way and simultaneously/at the same time.

UART can be configured in any of the above ways as per user need.

ADVANTAGES:

- It is two wired.
- No clock required.
- Error can be detected easily using parity bits.
- Simpler and cheaper compared to parallel comm.

DISADVANTAGES:

- Does not support multi-master or multi-slave
- Speed is limited.

Efficiency of UART Protocol is 60-70% only.

SPI PROTOCOL: This protocol is used for communication between MCU and SD Card/MMC card.

SPI stands for **Serial Peripheral Interface.**

- SPI is Full Duplex protocol.
- It is synchronous serial protocol.
- It is four wired.
- It has single Master and multiple Slaves.
- It is widely used for short distance communication.
- Maximum speed that SPI can communicate is 10Mbps.

Four wires of SPI are

1. Serial Clock Line (SCLK).
2. Master Out Slave In (MOSI).
3. Master In Slave Out (MISO).
4. Slave Select/Chip Select (SS/CS – Active Low Pin).

In SPI always master initiates the communication first by pulling the slave select low. It means it selects that particular slave for communicating and start generating the clock signal.

- To write data to slave, master transmits the data on MOSI line and stops generating the clock signal.
- To read data from the slave master first sends a read signal to the slave and slave responds with the data on MISO line (known data length) and master stops generating the clock signal.
- For unknown data length from the slave master first sends the read signal and slave responds with data length and again sends the data on MISO line and master generates the clock for that length of data after data being received the master stops generating the clock signal.

SPI CONFIGURATION: SPI protocol can be configured based upon the two parameters

1. Clock Polarity (CPOL)
2. Clock Phase (CPHASE)

Clock Polarity (CPOL): It is used to determine the idle state of the clock.

- ➢ CPOL = 0 it is non-inverted state of the clock, it means the idle state of the clock is 0/ low state of the clock is transmitted first/ active state of the clock is 1.
- ➢ CPOL = 1 it inverted state of the clock, it means the idle state of the clock is 1/ high state of the clock is transmitted first/ active state of the clock is 0.

Clock Phase (CPHASE): It determines when the data has to be toggled by the Tx and when the data has to be sampled by the Rx.

- ➢ CPHASE = 0 it means data is sampled at the leading edge of the clock.
- ➢ CPHASE = 1 it means data is sampled at the trailing edge of the clock.

Based upon the clock polarity and clock phase SPI has 4 modes of operation. i.e., MODE 0, MODE 1, MODE 2, MODE 3

Mode	CPOL	CPHASE	Comment
0	0	0	Active state of clock is 1 Data is sampled at leading edge
1	0	1	Active state of clock is 1 Data is sampled at trailing edge
2	1	0	Active state of clock is 0 Data is sampled at leading edge
3	1	1	Active state of clock is 0 Data is sampled at trailing edge

Advantages:

1. Full Duplex.
2. Higher throughput than I2C.
3. Not limited to 8bit.
4. Simple hardware interface.
5. Typically, lower power consumption.

Disadvantages:

1. More GPIO's required as the slaves increase.
2. No hardware flow control.
3. No Acknowledgement signal.

I2C PROTOCOL: **Two Wire Interface control** (TWI) between the MCU and Camera module can be **I2C** basically.

I2C stands for **Inter-Integrated Circuit** Protocol.

- I2C is Half-Duplex communication.
- I2C is synchronous serial communication.
- It is two wire communication.
- I2C can have multi-master and multi-slave.
- It is basically used for short distance and intra board communication.
- MSB is transmitted first in I2C.
- Data transfer in I2C can be made with different speeds i.e.,
  100Kbps - Standard Mode
  400Kbps - Fast Mode
  1Mbps - Fast Mode Plus
  3.4Mbps – High Speed Mode
  This above are for Bi-Directional
  But in Uni-Directional it can transmit up to 5Mbps i.e., Ultra-Fast Mode

- I2C two wire can be Serial Data Line (SDA) and SCL (Serial Clock Line). These two wires are open drain/open collector lines and both are pulled up using a pull up resistor and tied to V_{DD}
- Every I2C based devices have their base address/device address which are used by the master during the communication with slave device.
  That can be of 7bit / 10bit.
  If it is 7bit we can connect 2^7 i.e., 128 devices on the bus and if it is 10bit slave address we can connect 2^{10} i.e., 1024 devices on the I2C bus.

Data Format:

Start + Device Address(7/10bit) + Ack + R/W(1bit) + + Memory location + Ack(1bit) + Repeated start(1bit) + Data(8bit) + Ack + Data + Ack + …. + Stop

I2C has two modes basically

1. Master Transmit Mode/ Slave Receive Mode.
2. Master Receive Mode/ Slave Transmit Mode

- In I2C always master has to initiate the communication.

Start Condition: A high to low pulse on the SDA line when the SCL is high is known as start condition.
Stop Condition: A low to high pulse on the SDA line when the SCL is high is known as stop condition.
Acknowledgement: SDA line is pulled low for one complete clock cycle is known as acknowledgement.
Read/Write: For read this bit is logic 1 and for write this bit is logic 0.

Master Transmit Mode/Slave Receive Mode:

- In this mode the master first sends the start bit then all the slave devices on the bus become active.
- Then, master sends device address on the bus then the slave with that address is active, also sends ACK to master and remaining slave devices goes into sleep.
- Then master transmit write bit i.e., logic 0, then after master gets ACK, and then master sends the memory location where he wants to write the data and again slave responds with ACK if that memory location is available free space or else NO ACK is given.
- Master again sends repeated start to that device along with data, and slave responds with ACK for every byte received to slave. This continues until master sends data.
- Through out this process master continuously generates clock signal for every bit transmitted and received and it stops generate clock once it sends stop bit to slave.

Master Receive Mode/ Slave Transmit Mode:

- In this mode also the master first sends the start bit then all the slave devices on the bus become active.
- Then, master sends device address on the bus then the slave with that address is active, also sends ACK to master and remaining slave devices goes into sleep.

- Then master transmit read bit i.e., logic 1, then after master gets ACK, and then master sends the memory location from where he wants to read the data and again slave responds with ACK.
- Master again sends repeated start to that device this time slave responds with the data and master ACK's for every byte received by master. This continues until master gets enough data and sends NO ACK and sends stop bit.

I2C Terminology to be known:

1. **Arbitration.**
2. **Clock Stretching.**

Arbitration: This means as the I2C consists of multi-master the bus access cannot be given to all the master nodes at a time. So, which ever master transmits 0 first on the I2C SDA line gets the bus access and remaining master nodes wait for there turn to get the bus access. This phenomenon is known as Arbitration.

Clock Stretching: Slave pulls the SCL low after receiving a byte of data indicating that the it is not ready for communicating. So the master has to wait until SCL line goes high. This phenomenon is known as Clock Stretching.

Advantages:

1. Supports multi-master and multi-slave.
2. Less number of pins required.
3. Error handling is better due to ACK.
4. Due to clock stretching it can work well with both slow and fast IC's

Disadvantages:

1. Slow transmission speed due to frame overhead because of device address bits and ACK bits.
2. Half Duplex
3. Hardware complexity increases with multi-master and multi-slave.