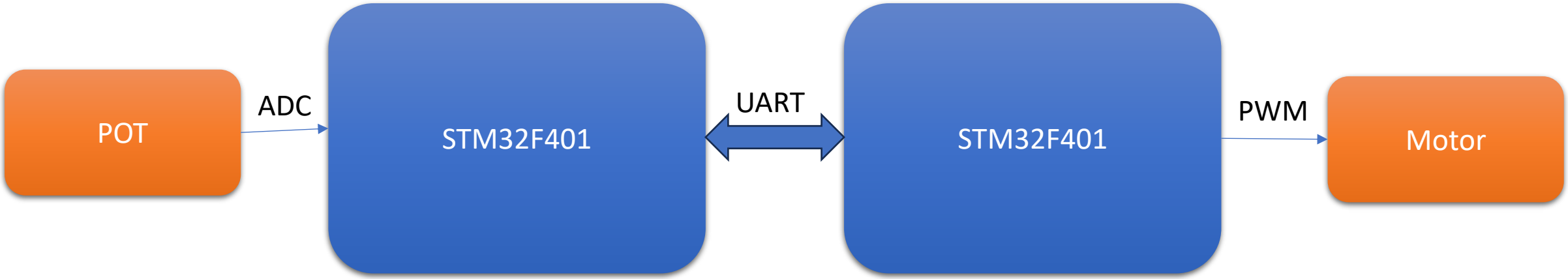
Module 12

Serial Peripheral Interface (SPI)

15th February 2025

Before Getting Onto SPI

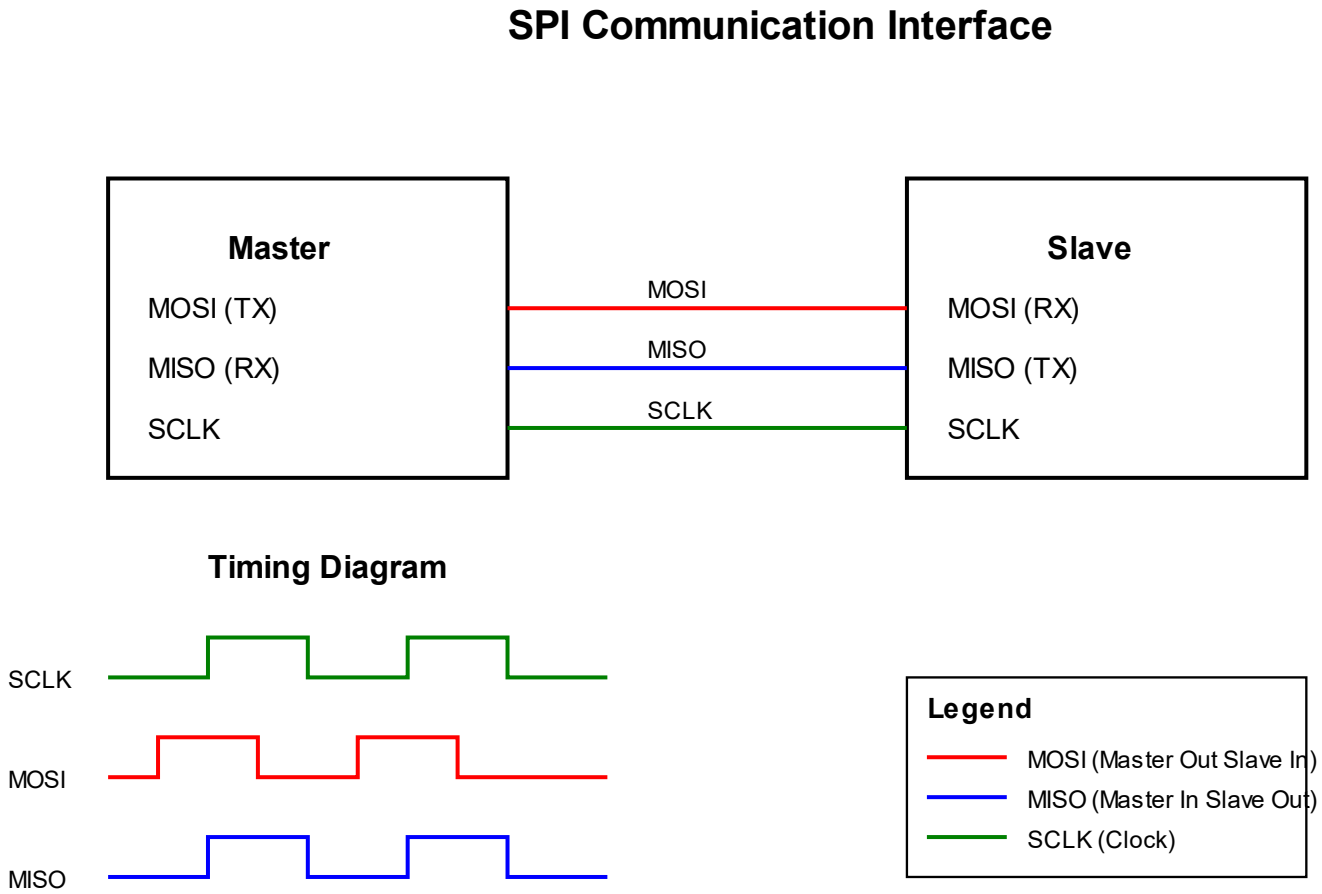


Time Duration For The Experiment – 45 Mins

Serial Peripheral Interface

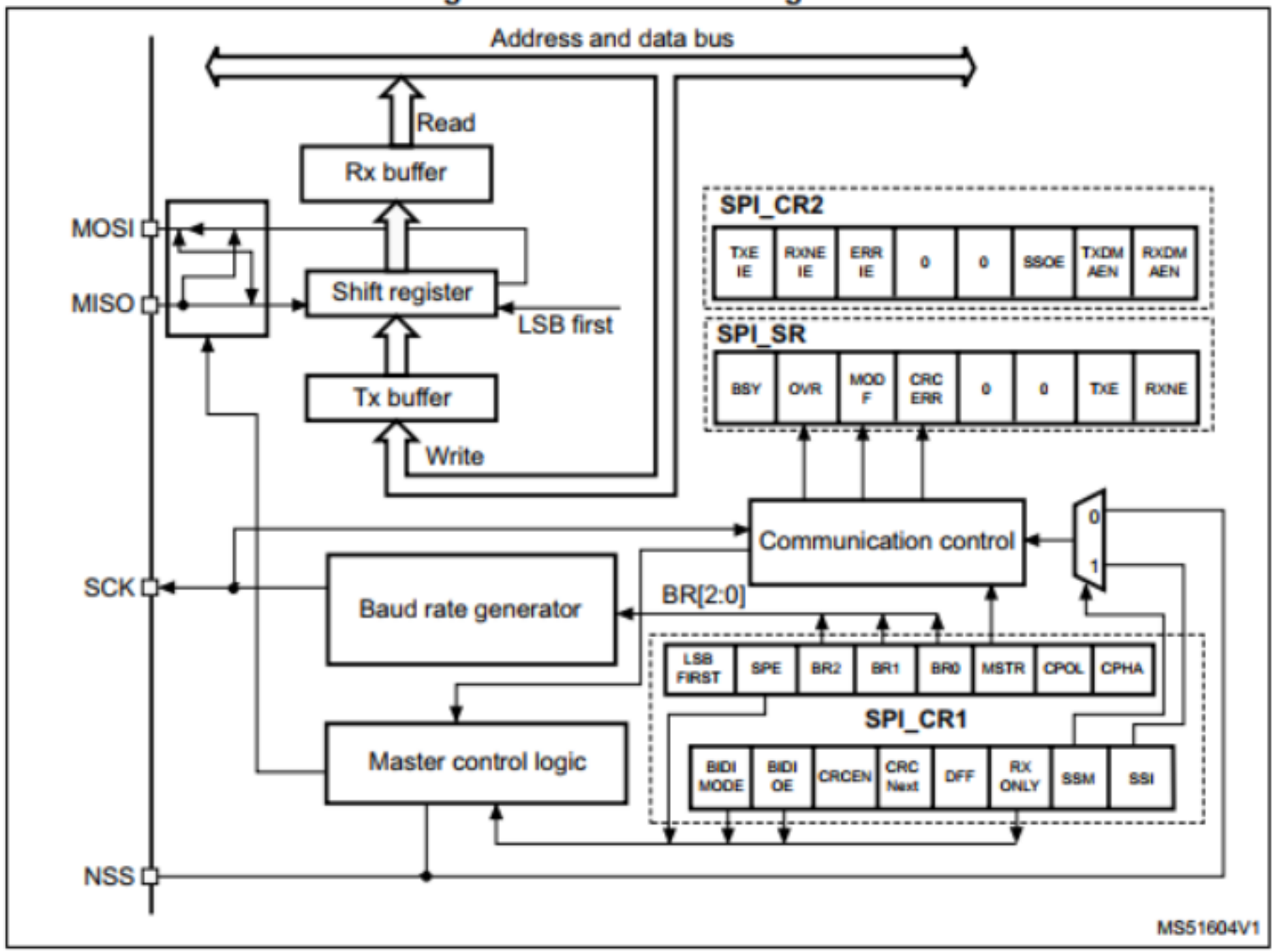
SPI (Serial Peripheral Interface) is a synchronous serial communication interface

- Specifies a 4-wire protocol for point-to-point communication
- Master-slave architecture where master controls all communications



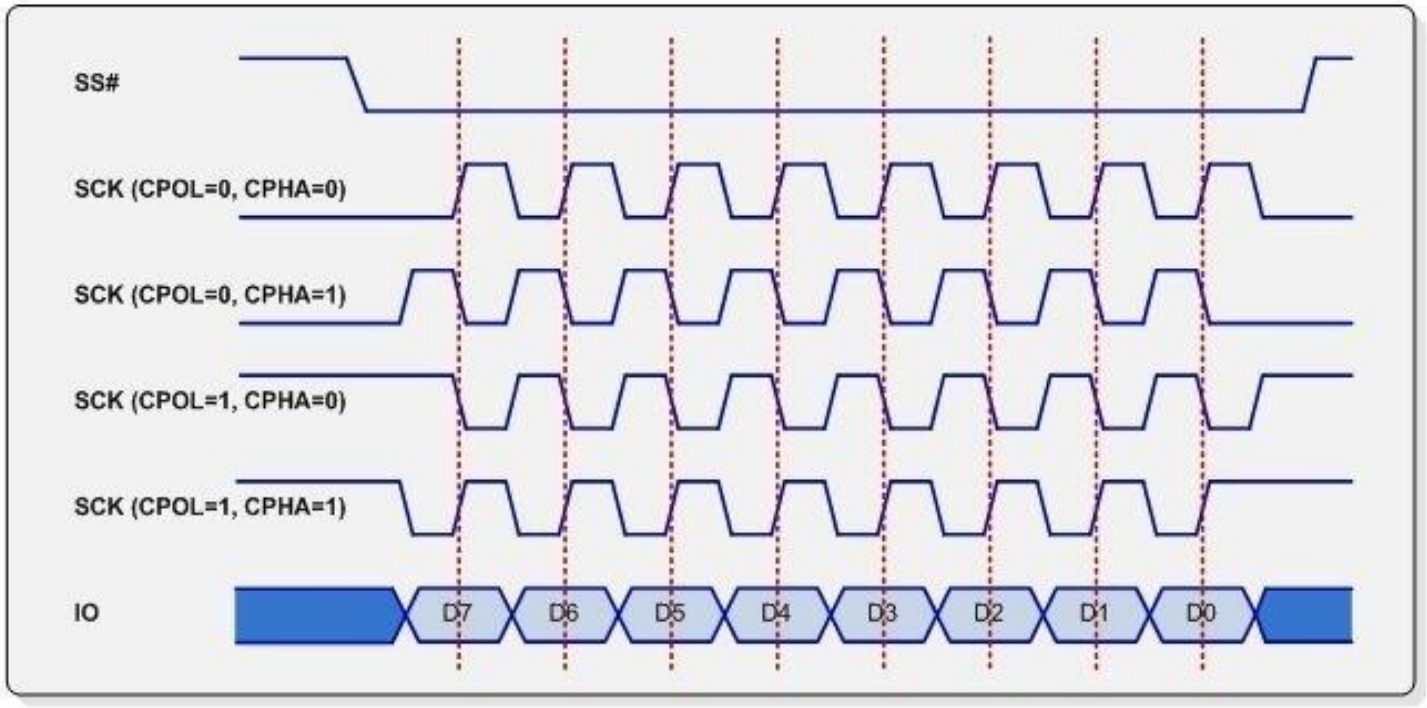
SPI in STM32

Figure 192. SPI block diagram



- Data is transmitted and received through the shift register using MOSI and MISO lines.
- The Rx buffer stores received data, while the Tx buffer holds data before transmission.
- SPI_CR1 and SPI_CR2 configure SPI settings, while SPI_SR indicates status and errors.
- The baud rate generator controls the speed of communication based on BR[2:0] settings.
- The NSS (Slave Select) pin manages multi-device communication by enabling/disabling slaves.
- Data transfer occurs synchronously, with the master generating the clock signal via SCK.

SPI Modes



Mode	CPOL	CPHA	Clock Idle State	Data Sampling
Mode 0	0	0	Low (0)	Capture on Rising Edge
Mode 1	0	1	Low (0)	Capture on Falling Edge
Mode 2	1	0	High (1)	Capture on Falling Edge
Mode 3	1	1	High (1)	Capture on Rising Edge

SPI Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	*LSB FIRST*	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Bit Name	Description	Use Case
15	**BIDIMODE**	**Bidirectional Data Mode** 1 = Half-duplex (single wire for Tx/Rx) 0 = Full-duplex (separate lines for Tx and Rx)	**Used when a device supports only a single data line (e.g., some sensors that use half-duplex SPI).**
14	**BIDIOE**	**Output Enable in Bidirectional Mode** 1 = Transmit-only 0 = Receive-only	**If the device is in half-duplex mode and should only transmit data, set this bit.**
13	**CRCEN**	**Enable CRC Calculation** 1 = CRC enabled 0 = CRC disabled	**Used in applications where data integrity is critical, such as industrial automation.**
12	**CRCNEXT**	**Transmit CRC Next** 1 = Send CRC after last data byte 0 = Normal operation	**Used when CRC is enabled and the master must transmit the CRC value at the end.**
11	**DFF**	**Data Frame Format** 1 = 16-bit data format 0 = 8-bit data format	**If the device communicates using 16-bit words (e.g., ADCs with 16-bit output), set this bit.**
10	**RXONLY**	**Receive-Only Mode** 1 = Only receive data (MISO active, MOSI ignored) 0 = Normal mode	**Used when SPI is reading data from a sensor (e.g., a temperature sensor) and doesn't need to send data.**

SPI Control Register

Thynk Loop

Bit	Bit Name	Description	Use Case
9	SSM	**Software NSS Management** **1 = Software controls NSS pin** **0 = Hardware controls NSS**	**Used when a software-controlled NSS signal is needed instead of hardware control.**
8	SSI	**Internal Slave Select** (Used with SSM=1) 1 = NSS internally high (Master mode) 0 = NSS internally low (Slave mode)	**Used when SPI is in master mode and doesn't have a physical NSS pin.**
7	LSBFIRST	**Data Transmission Order** 1 = LSB first 0 = MSB first	**Some devices require LSB-first communication (e.g., certain legacy EEPROMs).**
6	SPE	**Enable SPI Peripheral** 1 = SPI enabled 0 = SPI disabled	**SPI must be enabled before starting communication.**
5:3	BR[2:0]	**Baud Rate Control** 000 = fPCLK/2 001 = fPCLK/4 111 = fPCLK/256	**Used to adjust SPI speed according to the slave device's timing requirements.**
2	MSTR	**Master Selection** 1 = Master 0 = Slave	**Set this if the microcontroller should act as an SPI master.**
1	CPOL	**Clock Polarity** 1 = Clock idle HIGH 0 = Clock idle LOW	**Different SPI devices require different CPOL values; check the slave device’s datasheet.**
0	CPHA	**Clock Phase** 1 = Sample on second edge 0 = Sample on first edge	**Some devices require CPHA=1 for correct timing.**

SPI Control Register - 2

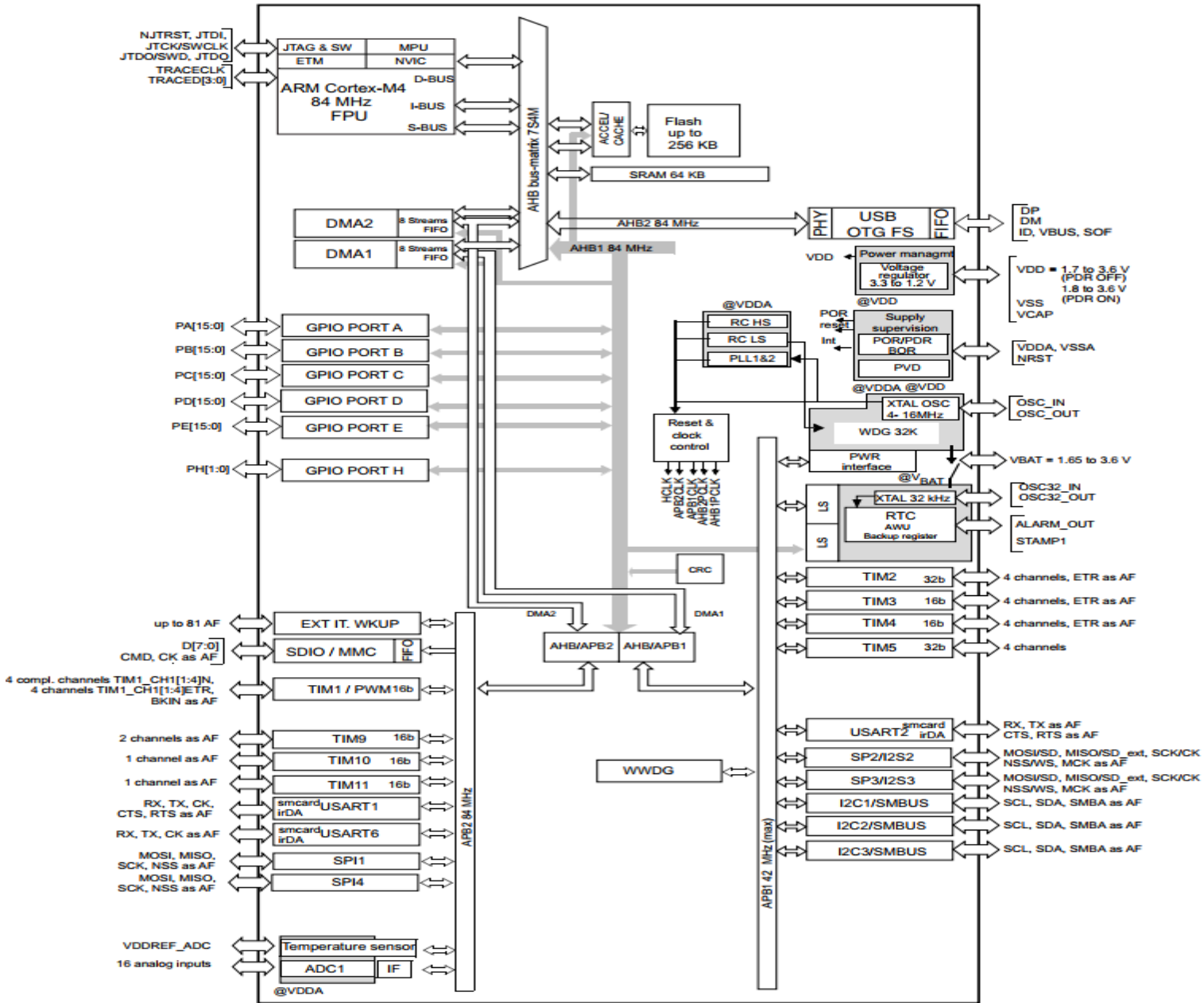
Thynk Loop

Bit	Bit Name	Description	Use Case
7	TXEIE	Transmit Buffer Empty Interrupt Enable 1 = Interrupt when TX buffer is empty 0 = No interrupt	Used in applications requiring non-blocking SPI transmission via interrupts.
6	RXNEIE	Receive Buffer Not Empty Interrupt Enable 1 = Interrupt when RX buffer has data 0 = No interrupt	Used to notify the processor when new data arrives (e.g., from a sensor).
5	ERRIE	Error Interrupt Enable 1 = Interrupt on SPI error 0 = No interrupt	Used when error handling is critical, e.g., in safety-critical systems.
4	Reserved	Always 0	Reserved for future use.
3	FRF	Frame Format 1 = TI mode 0 = Motorola SPI mode	Used when interfacing with devices that follow the TI protocol.
2	SSOE	Slave Select Output Enable 1 = NSS forced low in master mode 0 = Normal NSS behavior	Used in multi-slave setups where NSS must be actively controlled by the master.
1	TXDMAEN	Enable DMA for Transmission 1 = DMA enabled 0 = DMA disabled	Used in high-speed SPI communication to offload processing from the CPU.
0	RXDMAEN	Enable DMA for Reception 1 = DMA enabled 0 = DMA disabled	Used when receiving a continuous data stream from a sensor.

Serial Peripheral Interface

Bit	Bit Name	Description	Use Case
7	BSY	SPI Busy Flag1 = SPI is busy0 = SPI is idle	Check this flag before starting new communication.
6	OVR	Overrun Error Flag1 = Data overflowed0 = No error	If the SPI is too slow in reading data, this flag is set. Clear it to resume operation.
5	MODF	Mode Fault (Only in Master Mode)1 = NSS pulled low due to incorrect mode0 = No fault	Occurs when multiple masters drive the SPI bus incorrectly.
4	CRCERR	CRC Error1 = CRC mismatch0 = CRC correct	Used in applications where data integrity is critical.
2	TXE	Transmit Buffer Empty1 = TX buffer empty0 = Data still in TX buffer	Check before writing new data to avoid overwriting unsent data.
1	RXNE	Receive Buffer Not Empty1 = Data available0 = No data	Check this before reading data to avoid reading invalid values.

Baud Rate Generation



Baud Rate Generation

Thynk Loop

BR[2:0] Value	Prescaler	SPI Clock (f_{SPI})
000	$f_{PCLK}/2$	Fastest
001	$f_{PCLK}/4$	
010	$f_{PCLK}/8$	
011	$f_{PCLK}/16$	
100	$f_{PCLK}/32$	
101	$f_{PCLK}/64$	
110	$f_{PCLK}/128$	
111	$f_{PCLK}/256$	Slowest

SPI CONFIGURATION SEQUENCE

Thynk Loop

- Enable the SPI peripheral clock by setting the corresponding SPIxEN bit in the RCC register.
- Configure GPIO pins for SPI functions by setting them to Alternate Function mode and adjusting speed and pull settings.
- Configure NSS pin as GPIO output. Set the GPIO NSS pin as high.
- Set the SPI control registers by selecting baud rate, SPI mode (CPOL & CPHA), master or slave mode, and data format.
- Enable the SPI peripheral by setting the SPE bit in the SPI_CR1 register.
- Configure NSS pin as low.
- Transmit and receive data by checking the TXE and RXNE flags in SPI_SR and writing or reading from SPI_DR.
- Configure NSS Pin as high
- Disable the SPI peripheral by clearing the SPE bit in SPI_CR1 when communication is complete.