**CC3000 Notes**

CC3000 Supported clock rate for SPI = 0-16MHz, so set clock rate on MCU accordingly

(strangely, CC3000 host programming guide says 0-26MHz)

SPI on Atmega168 MCU

* Must set PRSPI (PRR’s bit 2) to 0 to enable SPI module
* Set the SPI Interrupt Enable bit (SPIE) in the SPCR Register to have an interrupt requested after each byte is sent/received
* PB5 (pin 19) = SCK (Master clock output) (DDB5 must be set properly for this to be output
* PB4 (pin 18) = MISO (master input)
* PB3 (pin 17) = MOSI (master output) (DDB3 must be set properly for this to be output)
* PB2 (pin 16) = SS (slave select) (DDB2 must be set properly for this to be output)
* Interrupt 0x0022 is SPI STC (serial transfer complete)
* Send byte by writing to it to SPDR after pulling SS low
* Read from SPDR reads byte that was received from slave during last byte xmit
* SPI uses the I/O clock, not the CPU clock

Master pulls SS low to select the slave.

Master and slave put data into their respective registers.

Master generates required clock cycles on SCK to interchange data.

After each data packet (series of bytes), master pulls SS high to synchronize slave.

|  |  |  |
| --- | --- | --- |
| **CPU Clock** | **Div** | **Output Clock** |
| 14745600 | 2 | 7372800 |
|  | 4 | 3686400 |
|  | 8 | 1843200 |
|  | 16 | 921600 |
|  | 32 | 460800 |
|  | 64 | 230400 |
|  | 128 | 115200 |

Code must:

* Set PRSPI of PRR to 0 to enable SPI module
* Set DDB3 (MOSI) and DDB5 (SCK) to 0 (making them output lines)
  + MISO is auto configured as input when MCU is configured as master
* Set SPE of SPCR (bit 6) to 1 to enable SPI
* Set DORD of SPCR to 0 for Most-Significant-Bit-first transmission
* Set MSTR of SPCR to 1 to set MCU as SPI Master
* Set CPOL and CPHA of SPCR to 0
* Set SPR1 and SPR0 of SPRC to 0 (fastest clock speed : 1/4th)
* Set SPIF of SPSR to 0 (no interrupt)
* Set SPI2X in SPSR to 1 (for double speed: 1/2)
* Write 0 to SS
* While bytes-to-write
  + Write a byte to SPDR (SPI Data register)
    - Starts SPI clock
    - SPI shifts bits into the slave
    - After one byte, clock is stopped
    - SPIF (SPI end-of-transmission flag) is set
    - Interrupt request if SPIE (SPI Interrupt Enable) in SPCR is set
* Write 1 to SS