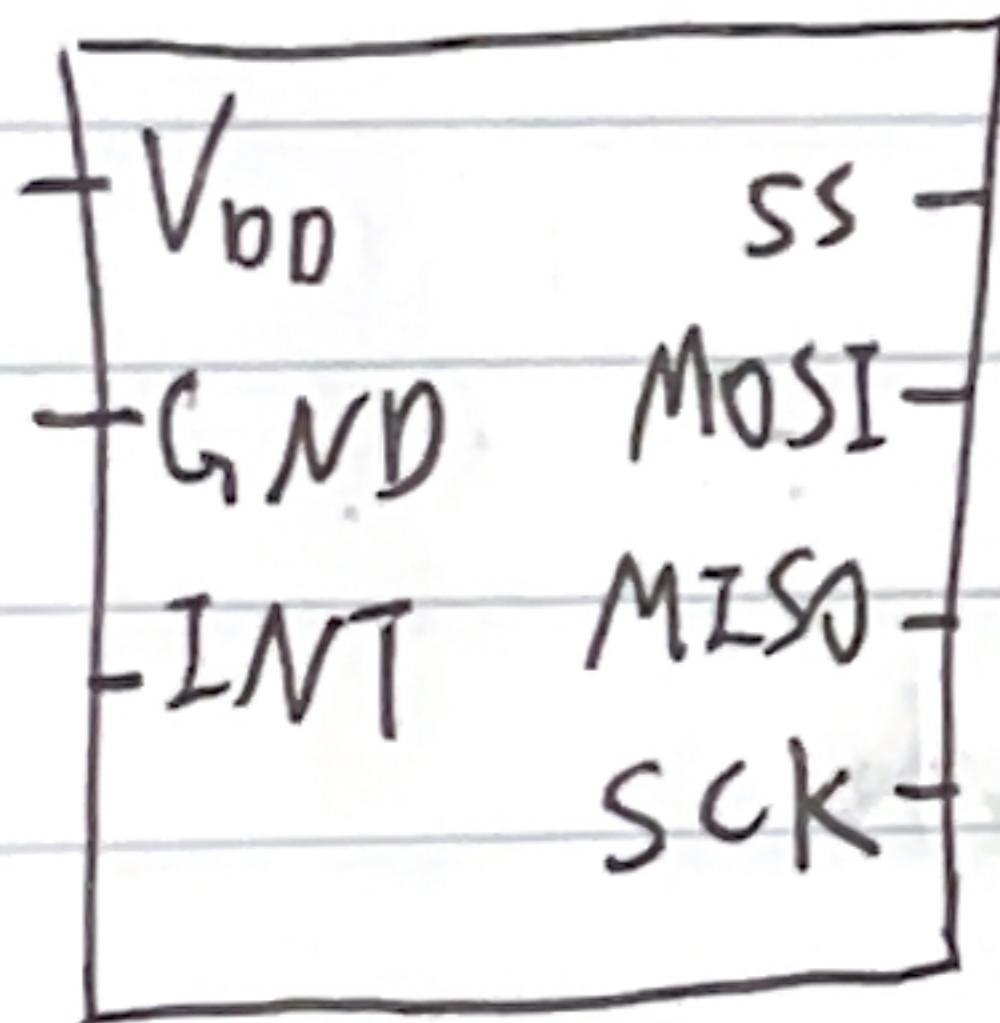


Question 2.



INT: Data-ready Interrupt.

It provides INT pin for interrupt. When new respiratory data is available, this will trigger the INT pin.

Register Mapping

R/W	Register Add	B7	B6	B5	B4	B3	B2	B1	B0	POF STATE
R	Status	0x00	DRDY	OVRF	ERRO					0x00

Bit7: DRDY: Data Ready flag

After a sample of respiratory rate is collected, the interrupt triggers.

Bit 6 : OVRF : overflow flag

When new respiratory data is generated before the previous data was read, this bit would be set to 1.

Bit 5 : ERRO : error flag

Any internal error condition happens, this bit is set to 1.

Bit 4:0 : Reserved.

Interrupt Enable (0x01)

R/W	Register	Add.	B7	B6	B5	B4	B3	B2	B1	B0	PUR STATE
R/W	Int-Enable 0x01		En-DRDY	En-OVRF	En-ERRO	\	\	\	\	\	0x00

Bit 7: En-DRDY : Enable data ready interrupt
when DRDY=1, enable the data ready interrupt.

Bit 6: En-OVRF : Enable Overflow interrupt
when OVRF=1.

Bit 5: En-ERRO : Enable Error interrupt
when ERRO=1.

Bit 4:0 : Reserved.

Data Read 0x02-0x03

R/W	Register Add	B7 B6 B5 B4 B3 B2 B1 B0	POR STATE
R/W	Data Read 0x02 High	DATA-H [7:0]	0x00
R/W	Data Read 0x03 Low	DATA-L [7:0]	0x00

Every sample of Data is an 16-bit unsigned integer.

In MCU, converting into decimal and multiplied by 0.1 will be respiratory rate in breaths per minute.

For example, if Data get is below,

ob 0 000 0 001 0 011 0 010

converts into hexadecimal is 0x0132

converts into decimal is 306

calculation: $306 \times 0.1 = 30.6$

respiratory rate: 30.6 bpm

Configuration 0x04

R/W Register Add B7 B6 B5 B4 B3 B2 B1 B0 POR STATE

R/W CONFIG 0x04 ENAB \ \ \ \ \ \ RESET 0x00

Bit 7 : Enable sensor
when set to 1

Bit 6:1 : Reserved

Bit 0 : Reset sensor
when set to 1

SPZ Communication

Mode 0 : CPOL=0, CPHA=0

- means data sampled on rising edge, clock idle low.

1. Write Single Register. eg. write ob10000000 to 0x04

SS

01001100 10000000

SCK

00101010 01010101

MOSI

MISO

2. Read Single Register eg. read ob10000000 at 0x00

SS

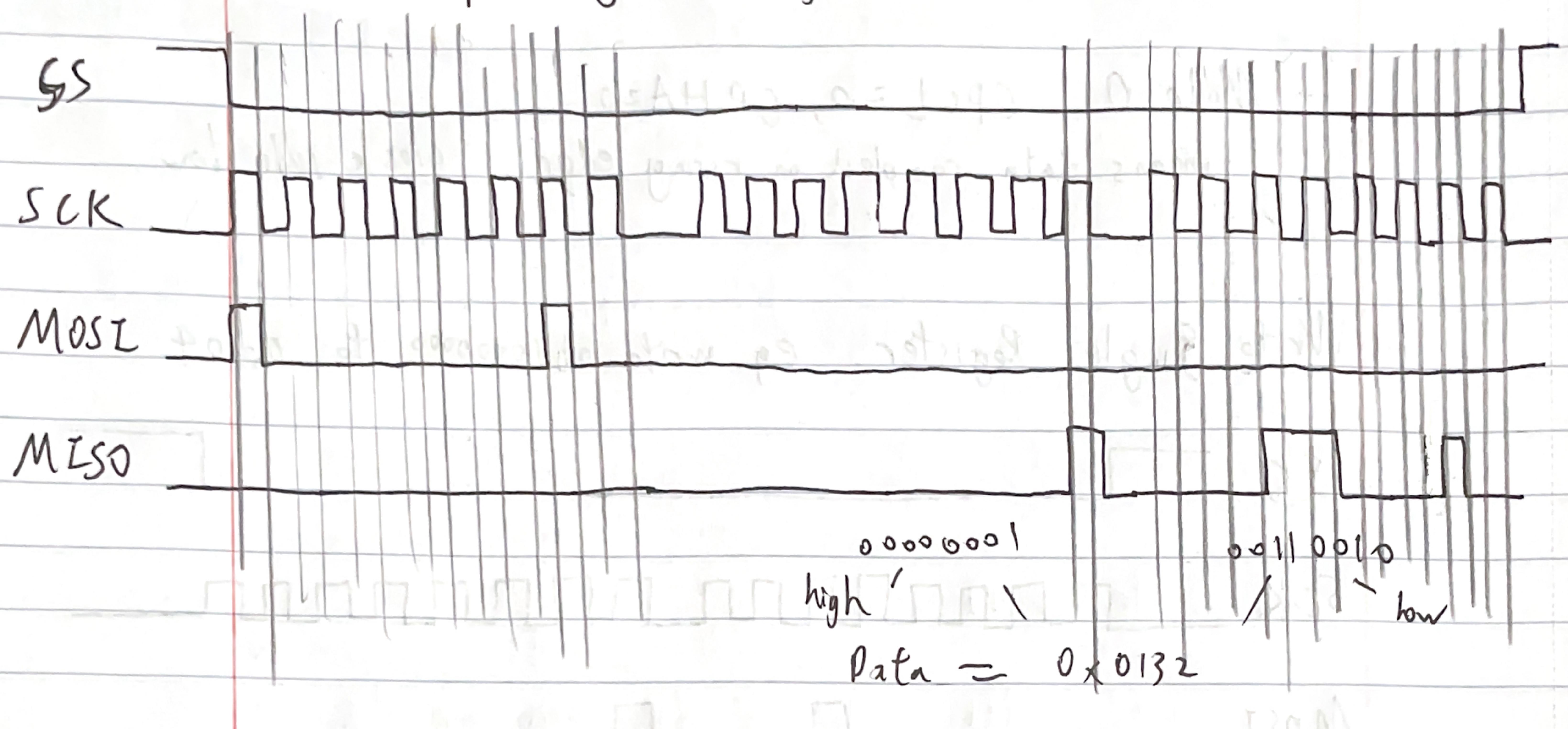
(RDY at status)

SCK

MOSI

MISO

3. Read Multiple Register eg. read 0x02-0x03 to get data.



Pseudo-code example

// 1. Configuration

```
SPI-init();           // choose SPI mode ,MSB, clk speed.  
Set-CS-pin-as-output();  
Set-CS-high();
```

// 2. enable sensor

```
Set-CS-low();  
SPI-write(0x04);          // address  
SPI-write(0x80);          // data.  
Set-CS-high();
```

// 3. Read data (Suppose data ready interrupt is called)

```
Set-CS-low();  
SPI-write(0x82);          // read from 0x02, first bit (as write).  
high-byte = SPI-write(0x00);  
low-byte = SPI-write(0x00);  
Set-CS-high();
```

// 4. data processing

```
rr_raw = (high-byte << 8) | low-byte;  
respiratory_rate = rr_raw / 10.0;
```