

Serial Communication

AS5132 Rotary Magnetic Position Sensor

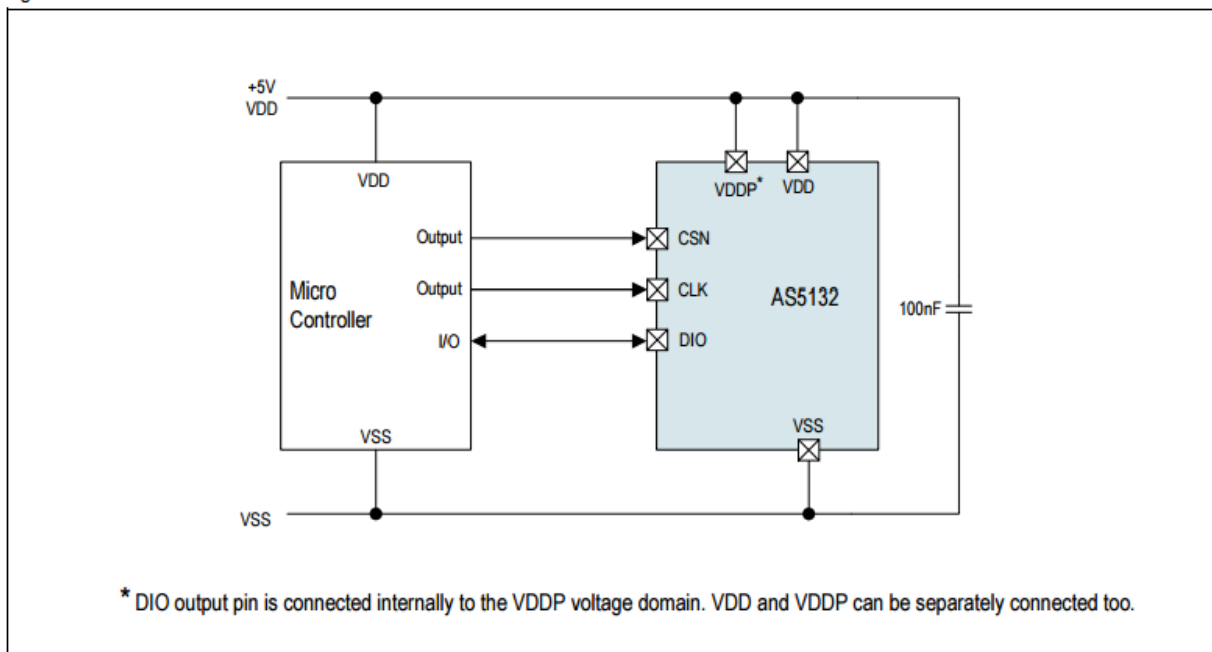
Stephen Dunn – 11/13/2015

The AS5132 is a rotary magnetic position sensor capable of measuring the absolute rotational angle of a magnetic field placed directly above the chip. In a typical application, this magnetic field is generated by a single dipole permanent magnet placed a small distance above the surface of the chip and allowed to rotate.

Description of the Electrical Interface

The AS5132 is capable of transmitting its angle measurement through several different mechanisms. The fastest and most precise of those mechanisms is through its 3-wire synchronous serial interface. The interface consists of a single bi-directional data line (DIO), a clock line (CLK), and a chip-select line (CSN). Additionally, the chip must be provided with 5 volts power and ground. As shown in the diagram below, additional capacitance (100nF or so) between the power and ground may be needed to reduce supply noise to the chip caused by various factors. It is important to note that the DIO data line is bi-directional meaning that bits can (and will) be driven on the line by both the microcontroller and the AS5132 sensor at various times. It will be important to be able to set the pin-mode of the IO pin on the microcontroller as an input or output pin accordingly so as not to connect two output pins together inadvertently. The diagram below gives a good overview of the required circuitry for communication.

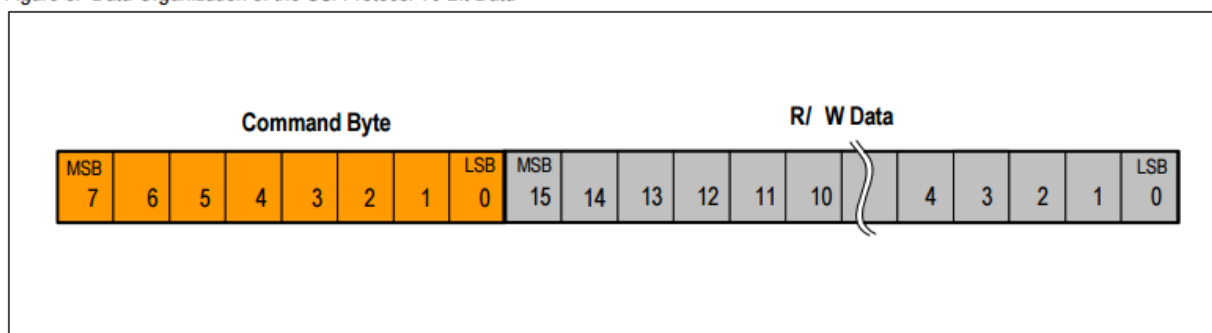
Figure 4. Read / Write Serial Data Transmission



Description of the Communication Protocol

The serial communications protocol consists of two distinct phases within a single communication frame. Each frame is 24 bits long with the first phase of the frame being an 8-bit command sent from the microcontroller. This command is followed up immediately by 16 data bits. The direction of these data bits depends on the particular command sent in the first 8 bits. The following diagram gives an overview of the bits in a frame.

Figure 5. Data Organization of the SSI Protocol 16-Bit Data



To begin a frame, the microcontroller pulls the chip select (CSN) line low. The AS5132 will immediately begin listening for data bits. The value on the DIO line is latched into the memory of the AS5132 on the rising edge of the CLK line (controlled by the micro controller). This is repeated 8 times for each of the 8 command bits.

If the command byte just sent represented a write command, the AS5132 continues latching in data bits (starting with the MSB of the data) on the very next rising clock edge, continuing until all 16 bits have been latched in.

If the command byte just sent represented a read command, the AS5132 will take over the operation of the data line, switching it from an input to an output. It is important that the microcontroller consider this transition and change the IO pin from an output to an input before the falling edge of the 8th clock pulse. On the falling edge of the 8th clock, the AS5132 will begin driving data bits onto the DIO line starting with the most significant bit of the 16 bit data word. On the falling edge of each clock (still controlled by the microcontroller) the AS5132 will put the next bit on the data line.

After the falling edge of the 24th clock pulse, the microcontroller pulls the chip select pin high to complete the data frame. The AS5132 will now wait for the next falling edge of the chip select line to begin another transmission frame.

The timing diagrams on the next page show the details of the protocol described above in both read and write mode.

Figure 6. SSI Timing in Write Mode

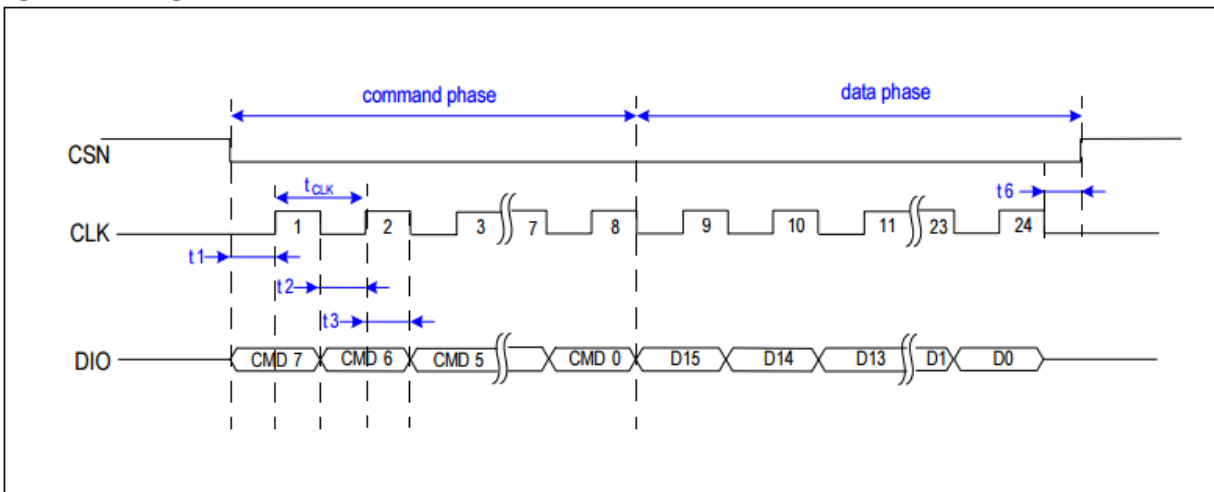
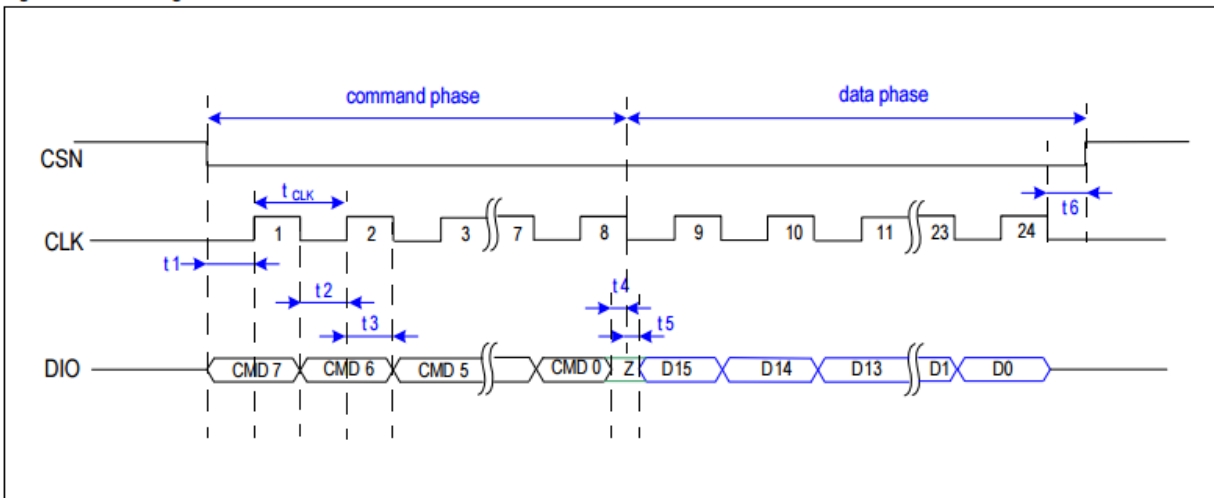


Figure 7. SSI Timing in Read Mode



Available Commands

The AS5132 is capable of receiving a number of commands to both read and write data to and from the sensor. Below is a selection of some of the most useful commands for general use. Two read commands are available:

RD_ANGLE: 0000.0000

After issuing this command byte, the sensor will respond with 16 bits containing the angle measurement in the first 9 bits, the "Lock ADC" flag in the 10th bit, and field strength value in bits 11 through 15. The 16th bit is a parity bit (odd parity).

RD_MT_COUNTER: 0000.0100

After issuing this command byte, the sensor responds with 16 bits containing the multi-turn counter value in the first 9 bits and a parity bit in the last bit. All other bits can be ignored.

To reset the multi-turn counter, the following write command is quite useful:

SET_MT_COUNTER: 0001.0100

After issuing this command, the sensor expects the 9 bits of the multi-turn counter value to be given in the first 9 bits of the data. The remaining 7 bits will be ignored.

The following table gives a summary of the above commands:

Command Name	Command Data	Access Mode	MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
SET MT COUNTER	0001_0100	write	MT - COUNTER <8:0>															
RD MT COUNTER	0000_0100	read	MT - COUNTER <8:0>									EZ ERR						P
RD_ANGLE	0000_0000	read	ANGLE <8:0>									LOCK ADC	AGC <5:1>					P

Sample Code

Using an ARM Cortex M4 and the TIVA Ware libraries from Texas Instruments

The configuration and communication with the AS5132 begins with a proper setup routine. All connected pins need to be enabled for GPIO, their direction needs to be set, and they need to be configured to a good initial condition.

```
ROM_GPIOPinTypeGPIOOutput(mPort, mPinCS);
ROM_GPIOPinTypeGPIOOutput(mPort, mPinCLK);
ROM_GPIOPinTypeGPIOInput(mPort, mPinDIO);
ROM_GPIOPinWrite(mPort, mPinCS, ALL_HIGH);
ROM_GPIOPinWrite(mPort, mPinCLK, ALL_LOW);
```

A transmission frame begins by pulling the chip select pin low. In this example, we will be executing a read command.

```
ROM_GPIOPinWrite(mPort, mPinCS, ALL_LOW);
```

Next, the first 7 bits of the command can be sent to the sensor. The 8th bit requires some special handling because the data line will switch from an output to an input.

```
ROM_GPIOPinTypeGPIOOutput(mPort, mPinDIO); // set DIO pin to an output.
for(int i = 0; i < 7; i++) {
    // write data bit, MSB first
    ROM_GPIOPinWrite(mPort, mPinDIO, (command & 0x80) >> 7);
    command = (command << 1);

    // pulse the clock to latch the bit in the AS5132
    ROM_GPIOPinWrite(mPort, mPinCLK, ALL_HIGH);
    ROM_GPIOPinWrite(mPort, mPinCLK, ALL_LOW);
}
```

Special handling for the 8th bit and the transition from output to input:

```
// write last data bit
ROM_GPIOPinWrite(mPort, mPinDIO, (command & 0x80) >> 7);
command = (command << 1);

// clock it in on the rising edge
ROM_GPIOPinWrite(mPort, mPinCLK, ALL_HIGH);

// before the falling edge of the clock, change the pin mode
ROM_GPIOPinTypeGPIOInput(mPort, mPinDIO);

// now let the clock fall.
// the AS5132 will immediately begin outputting data on the DIO line
ROM_GPIOPinWrite(mPort, mPinCLK, ALL_LOW);
```

Next, the 16 data bits are read from the sensor one at a time and shifted into their storage location in memory:

```
// 16 data bits
for(int i = 0; i < 16; i++) {
    // read data bit (MSB first)
    tmp = ROM_GPIOPinRead(mPort, mPinDIO);

    tmp = tmp >> 4; // get the bit in the ones position
    data = data | (tmp << (15 - i)); // shift it into the result

    // clock in the next bit.
    ROM_GPIOPinWrite(mPort, mPinCLK, ALL_HIGH);
    ROM_GPIOPinWrite(mPort, mPinCLK, ALL_LOW);
}
```

Finally, the chip select pin is pulled high, signifying the end of the transmission frame. The AS5132 will wait for the chip select line to be pulled low to begin another transmission.

```
ROM_GPIOPinWrite(mPort, mPinCS, ALL_HIGH);
```

References

All information and Images taken from the AS5132 Data Sheet from AMS:

http://ams.com/kor/content/download/254823/1001597/file/AS5132_Datasheet_EN_v5.pdf