Santa Barbara Imaging Group (SBIG)
CFW-9/v2 Filter Wheel
Interface and software protocol
Keith Rickard 28 July 2020

Introduction

The CFW-9 filter connects to the I2C AUX port of the USB version of the ST-7/8/9/10/200 series of CCD cameras. It replaced the CFW-8 filter wheel, which connected to the AO7/CFW8/SCOPE port, after SBIG were unable to source electronic parts for the CFW-8. It was released in 2006. Both filter wheels have five 1.25" slots

Its design, as with the CFW-8, allows it to be screwed to the front of the camera, matching the camera body's shape and making it look like a continuation of the camera body.

Why this Article?

Although the SBIG ST-7/8/9/10/2000 series are great cameras, technology has moved on. Upgrading to new astronomical kit is expensive. Presently, the CFW-9 can only work when connected to the ST-x camera body and therefore rendered unusable with other CCDs. It is therefore desirable to redeploy it and use it independently with other cameras.

SBIG's other filter wheel offerings for use with the ST-x cameras are the CFW-8 and CFW-10, which hold 5 and 10 filters respectively. SBIG have released the interface and protocol details for controlling the CFW-8 and the software protocol for the CFW-10 through its own RS232 interface. It has never released the same for the CFW-9. The CFW-8, CFW-9 and CFW-10 are all now discontinued.

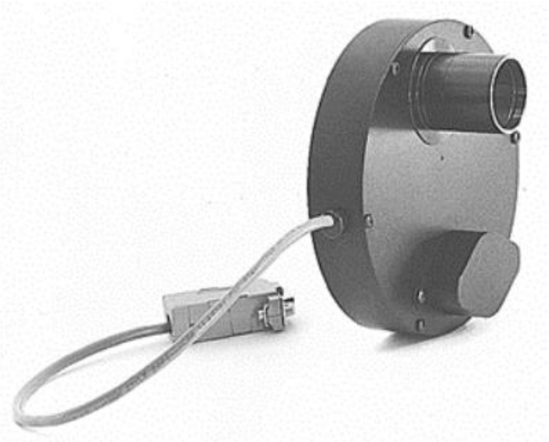
Diffraction Limited (in Canada) now owns SBIG and when asked in July 2020 for these details its reply was that this "…information is proprietary and confidential. We do not support operation via third party hardware and doing so may void your warranty." A very unhelpful response and I doubt any CFW-9 out there is still under warranty!

So, I went about working out this proprietary and confidential information myself and my findings are given in this article. With this information, any other camera can be connected to the T-thread hole of the filter wheel and an ASCOM driver can be developed to control it using, perhaps, an Arduino microcontroller or similar.

WARNING: These are my findings and I will not be responsible for any damage, harm or loss caused by any inaccuracies when using them.

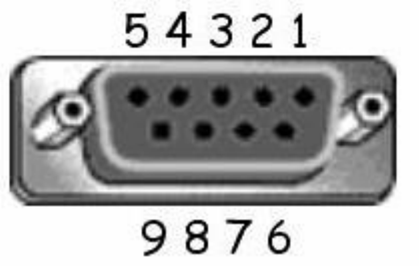
USER BEWARE!

The Hardware



The CFW-9 filter wheel has a short cable coming out of its side with an in-line female DB-9 socket at the end of it. Its primary purpose is to connect to the USB SBIG ST-7/8/9/10/2000 series CCD camera body's I2C/AUX port so that power and communication lines are facilitated. The I2C interface is used to control the CFW-9.

This diagram shows which of the DB-9 socket's pins are used.



Pin 3	SCL (Serial Clock)
Pin 4	SDA (Serial Data)
Pin 5:	Ground
Pin 8	+12V DC (max 250mA)

The shell is connected to the cable shielding (chassis ground).

The SCL and SDA are the I2C (TWI) signal lines and are connected to pull-up resistors. The I2C clock speed is a standard 100KHz

IMPORTANT: the SCL and SDA lines operate at +3.3V, not +5V like other I2C set-ups. Using the wrong voltage could damage the CFW-9.

The +12V DC line provides power to the CFW-9. When the CFW-9 is idle, only 10mA of current is drawn. When its motor is turning, the current seems to be a maximum 250mA.

When the CFW-9 is powered up, the filter wheel always turns to find its home position. When moving, the selected position is always reached within four seconds.

The Command Protocol

The filter wheel is an I2C slave device with 52h as its address. There are only two commands which are sent using the SCL and SDA signal lines.

Move to Filter

The following three bytes are sent to the CFW-9:

A4h 10h XXh

Where XXh is the selected filter position 01h through to 05h. If 00h is selected then the filter wheel will go to home position, similar to selecting 01h – the difference is that the wheel will still move even if at the home position.

A4h is 52h shifted left one bit with bit 0 then reset to 0 (write flag).

Get Status Byte

To get the CFW-9's Status byte, the following byte is sent to the CFW-9:

A5h

A5h is a 52h shifted left one bit with bit 0 then set to 1 (read flag).

The CFW-9 will then respond with one byte which is the status byte. Generally, the bits of the status byte represent the following.

Bit No.	Function	Details
7	x	Not used(?)
6	Time out error	Set if motion of filter wheel is not detected in the last 1-2 seconds. It will remain set even if motion is detected later. It will be reset when the wheel moves to another position.
5	x	Not used(?)
4	Moving state	1 = moving (not yet arrived), 0 = stopped at filter position.
0 - 3	Filter number, etc	These four bits represent the filter position number when not moving. If moving then other values are given to indicate the state of the motion of the wheel. See below for more details

The Status byte has been found to have these values and apparent reasons:

01h	Wheel not moving and at position 1.
02h	Wheel not moving and at position 2.
03h	Wheel not moving at position 3.
04h	Wheel not moving at position 4.

05h	Wheel not moving at position 5.
0Fh	Wheel already at selected position. Value is given when the wheel is told to move to the position that it is already at. **Beware –** this value is always returned when the CFW-9 has been asked to go to the current position meaning information of precisely where it is can no longer be retrieved. Only moving the wheel to a new position, or home, will reset this.
1Fh	Wheel moving to selected position.
10h	Wheel moving to home position. This value is only returned when the CFW-9 is initialising at power-up or if the move command A4h 10h 00h has been sent (i.e. go to home position).
11h	Wheel nearing home position (between 5 and 1).
13h	Wheel nearing position 2 (between 1 and 2).
14h	Wheel nearing position 3 (between 2 and 3).
15h	Wheel nearing position 4 (between 3 and 4).
5Fh	Wheel was sent moving but the optical sensor has not detected its motion within 1-2 seconds. The wheel may be stuck or struggling to turn.
41h	Wheel at position 1, not moving and took excessive time getting there.
42h	Wheel at position 2, not moving and took excessive time getting there.
43h	Wheel at position 3, not moving and took excessive time getting there.
44h	Wheel at position 4, not moving and took excessive time getting there.
45h	Wheel at position 5, not moving and took excessive time getting there.

The above findings may not be a complete understanding of the Status byte values. Curiously, there appears to be no values for when the wheel is almost at position 1 or 5.

Conclusion

The CFW-9 interface is fairly straight forward, though the control system needs careful consideration. It would need to supply +12V capable of at least 250mA current draw and be able use +3.3V on the SDA and SCL signal lines.

The command protocol is also simple. At a minimum, the *Move to Filter* is used to go to a filter position and assume the wheel will be there after four seconds. Testing bits 6 and 4 of the Status byte will provide error detection. Bits 0 to 3 will confirm the wheel position.

The CFW-10 RS232 Interface

I have included the SBIG release notes (21 September 2004) of the CFW-10 filter wheel. The CFW-10 also has an I2C interface and its RS232 command structure show some similarities with my CFW-9 findings.

Introduction

SBIG's CFW-10 ten position filter wheel can be controlled both directly by an SBIG camera with an IIC port, or by a computer through an RS-232 connection. SBIG provides a Visual Basic utility for control, or the user may wish to write his own control program. This memo summarizes the control of the CFW-10.

Basic Operation

When the CFW-10 is powered up is automatically rotates the filter wheel while watching for the pair of holes in the carrousel that indicate the home filter. It will always stop the second time it views them, and it will stop centered on the home filter. From then on the filter only moves one direction – it never reverses. This is done to improve the repeatability of each filter position. You can command any filter you want, but the wheel only rotates one way. The max time to move one full rotation is about 8 seconds.

Command Structure

The RS-232 communication takes place at 9600 baud. The data is 8 bits, no parity, and 1 stop bit. All command packets between PC and CFW-10 (in either direction) are 6 bytes long. The CFW-10 only responds to inquiries; the PC is the master.

The 6 bytes have the following designations:

Byte 1) Framing Byte: always a hex A5
Byte 2) Number of data bytes to follow: always a hex 3
Byte 3) Command byte: value depends on desired response
Byte 4) Low byte of parameter value (parameter is an unsigned integer)
Byte 5) High Byte of parameter value
Byte 6) check byte: value is modulus (lower 8 bits) of sum of other 5 bytes

The commands supported are described individually.

Calibrate:

The command is a hex 10 (decimal 16). The parameter value is zero. The CFW-10 responds with a single byte, a hex value of 6. The response is sent immediately, even though the move is still in process.

Move to Filter:

The command is a hex 11 (decimal 17). The low byte of the parameter value contains the filter position desired, from a 1 to a 10. A commanded position of zero results in a move to position 1. Any commanded position greater than 10 results in a move to position 10. The CFW-10 responds with a single byte, a hex value of 6. The response is sent immediately, even though the move is still in process.

Report Status Byte:

The command to read one of the sixteen status bytes is a hex 2. The low byte of the parameter value contains the byte desired. The CFW-10 will respond with a 6 byte string where the first byte is a hex A5, the second byte is the number of the status byte requested, the third byte is zero, and the fourth byte is the value of the status byte, if the status byte number requested was legitimate (0 through 15 decimal), or a 255 if it was outside of this range. Byte five will be a hex 40 (decimal 64). The meaning of the status bytes currently assigned are:

```
        InternalData[0] = Status;
        // lower 4 bits contain filter postition
        // bit 7 indicates IIC error if set
        // bit 6 indicates motor time out (excessive move)
        // bit 5 indicates max is only 10 units higher than min
        // bit 4 indicates motor is moving
        InternalData[1] = Filter;
        InternalData[2] = CalMax;
        InternalData[3] = CalMin;
        InternalData[4] = Threshold;
        InternalData[5] = Move % 256;
        InternalData[6] = Move / 256;
        InternalData[7] = IICAddress;
        InternalData[8] = IICData;
        InternalData[15] = Version
        ;
```

Only status bytes 0,1, and 15 have useful meaning for RS-232 inquiries. The first CFW-10 version shipped was hex 10 (decimal 16). One should poll status byte zero to determine if the motor has stopped moving after a calibrate or move to filter position command.

Report Calibration Data:

When the CFW-8 calibrates it records 2000 (0 thru 1999) sequential values of the opto-sensor reading, one per each step. The opto-sensor value goes high when it has a clear opening through one of the openings around the wheel. This data is useful in testing the filter wheel for proper operation. The command to read a calibration data point is a hex A5, followed by a hex 3, and then another hex 3. The fourth byte is the low byte of the location in the array for which cal data is desired, and the fifth byte is the high byte of the location desired. The sixth byte is the check digit. The data returned is a hex A5 followed by the low byte position echoed, the high byte position echoed, and the fourth byte containing the value of that calibration data point (range 0 thru 255). The fifth byte is zero, and the sixth byte the check digit. The user really does not need this information.

Change Baud Rate:

For some types of communications, such as downloading calibration data, 9600 baud is inconveniently slow. One can command the CFW-10 to switch to 115200 baud by sending a command hex 8 (decimal 8). The parameter value is not used. The CFW-10 responds with a single byte, a hex value of 6. The baud rate goes to the higher rate AFTER the response.

Miscellaneous Information:

The CFW-10 is designed to operate from unregulated 12 volts DC (11 to 14). It draws about 250 mA when moving, and 10 mA when stopped. The opto-sensor is well shielded, and turned off when the wheel is not moving.