

Temperature Control API in QHYCCD SDK

There is two basic control mode for temperature control. One is the Auto Mode. Another is Manual Mode.

Manual Mode

Camera is not working in auto temperature regular. Just set a PWM value of the TEC controller in camera. The PWM range is 0-255. 0 is mininum TEC power. 255 is maxium TEC power. Please note in this mode the temperature is not regulated, only the TEC power is fixed accroding the input value. The sensor temperature may change when enviroment temperature is changed. Normally you can use this function to do something like power ON/OFF sequencing. And also you can use this to do slow warm/slow cooling to prevent the sensor temperature changed too fast .

How to Eneter the Manual Mode:

Just use the CONTROL_ID of CONTROL_MANUALPWM in SetQHYCCDParam API. After you call this, the temperature control mode will switch to Manual Mode if previous is working in Auto Mode and then set the PWM vaule. And if the previous is the Manual Mode already, it will just set the PWM value.

```
//Enter Manual Mode and set the PWM value to 160 SetQHYCCDParam(h, CONTROL_MANUALPWM, 160);
```

In the Manual Mode, You can get the temperature, current PWM value by the CONTROL_ID of CONTROL_CURTEMP CONTROL_CURPWM. You can call them at anytime.

```
double currentTemp,currentPWM; currentTemp = GetQHYCCDParam(h, CONTROL_CURTEMP);  
currentPWM = GetQHYCCDParam(h, CONTROL_CURPWM);
```

Auto Mode

Camera will do the auto temperature regular automaticly. But please note for some old QHYCCD camera ,like QHY8L/10/11/12/15/16 etc. The auto temperature regular loop is not in the camera but in the SDK. **You need to call a API every one seconds to let this loop running.** The API can be call with the SetQHYCCDParam and the CONTROL_ID is CONTROL_COOLER. And after you call this API, The temperature control mode will switch to **Auto Mode**

```
//Enter Auto Mode and set the target temperature to -10.0C  
SetQHYCCDParam(h,CONTROL_COOLER,-10.0);
```

In Auto Mode, you can use the CONTROL_ID of CONTROL_CURTEMP CONTROL_CURPWM in GetQHYCCDParam to get the current Temerature and PWM value. You can call them in anytime. But please do not use the CONTROL_ID of CONTROL_MANUALPWM in SetQHYCCDParam API. Because it will cause switching to the manual mode.

There is another API can get the target temperature from the SDK. The CONTROL_ID of CONTROL_COOLER in GetQHYCCDParam. It will return the current target temperature you set into the SDK before.

```
//Get the current target temperature double t_Temp; t_Temp = GetQHYCCDParam(h,
```

CONTROL_COOLER);

Demo Codes

In actual host software, it always read the temperature, pwm value on each second or one second in a timer. And what we suggest something like this is like this.

```
void Timer_Event(){ if(automode) SetQHYCCDParam(h,CONTROL_COOLER,TargetTemp);
currentTemp = GetQHYCCDParam(h, CONTROL_CURTEMP); currentPWM = GetQHYCCDParam(h,
CONTROL_CURPWM); t_Temp = GetQHYCCDParam(h, CONTROL_COOLER); }
else{ SetQHYCCDParam(h, CONTROL_MANUALPWM, TargetPWM); currentTemp =
GetQHYCCDParam(h, CONTROL_CURTEMP); currentPWM = GetQHYCCDParam(h,
CONTROL_CURPWM); t_Temp = GetQHYCCDParam(h, CONTROL_COOLER); } }
```

How to Know if supported

some of QHYCCD camera does not support the temperature operation since they are uncooled camera , like the QHY5II series, QHY5III series uncooled camera. So that before you call the temperature API. You need to use the IsQHYCCDControlAvailable API to check if the camera supports it.

```
//if ret is QHYCCD_SUCCESS, it means supported. if ret is QHYCCD_ERROR it means not supported
uint32_t ret; //check if camera support to get/set current temperature ret =
IsQHYCCDControlAvailable(h,CONTROL_CURTEMP); //check if camera support to get/set current PWM
value ret = IsQHYCCDControlAvailable(h,CONTROL_CURPWM); //check if camera support to set the
AutoMode and get the targetTemp ret = IsQHYCCDControlAvailable(h,CONTROL_COOLER); //check if
camera support to set the Manual Mode ret = IsQHYCCDControlAvailable(h,CONTROL_MANUALPWM);
```

Other Parameters

The **CAM_CHIPTEMPTEMPERATURESENSOR_INTERFACE** is to readout the temperature from the temperature sensor on the cmos chip. Some cmos sensor has on-chip sensor so it can be readout here. For example. QHY5LII has such a onchip sensor. But since this sensor output is not so precise . So normally we do not use it.

The **CAM_TECOVERPROTECT_INTERFACE** is to set the TEC to 70% of max power to prevent the TEC working in too high voltage when the 12V input voltage become more than 13V (like use some battery, when battery is full) . But I have checked the whole project and looks we have not active this function in all cameras in current version.