

Software development kit for a compact cryo-refrigerator

J Gardiner, J Hamilton, J Lawton, K Knight, A Wilson and S Spagna

Quantum Design Inc.
10307 Pacific Center Court
San Diego, CA 92121, USA

Abstract. This paper introduces a Software Development Kit (SDK) that enables the creation of custom software applications that automate the control of a cryo-refrigerator (Quantum Design model GA-1) in third party instruments. A remote interface allows real time tracking and logging of critical system diagnostics such as pressures, temperatures, valve states and run modes. The helium compressor scroll capsule speed and Gifford-McMahon (G-M) cold head speed can be manually adjusted over a serial communication line via a CAN interface. This configuration optimizes cooling power, while reducing wear on moving components thus extending service life. Additionally, a proportional speed control mode allows for automated throttling of speeds based on temperature or pressure feedback from a 3rd party device. Warm up and cool down modes allow 1st and 2nd stage temperatures to be adjusted without the use of external heaters.

1. Introduction

Advanced cryo-refrigerator systems with enhanced functionalities such as the GA-1 cryo-refrigerator and the HAC900S variable speed compressor [1-3] are ideal for integration in third party instruments. This enables the user to have control of both main external devices as well as other peripheral devices (such as a temperature controller). The HAC 900S compressor supports a full-featured Control Area Network (CAN/CANopen) bus interface [4] that fully integrates with the other modules such as the measurement options used in the VersaLab™ Physical Property Measurement System [5]. The CANopen bus interface in our cryogenic systems has several benefits: non-critical cables and connector impedance requirements, good hardware support at the chip level, excellent bus arbitration and error checking, and adequate bandwidth for cryogenic laboratory systems. In addition, using a permanent “CANopen Loader” operating system, we can provide updated firmware via the CANopen bus, needing only a PC connected to the internet.

In its simplest embodiment, a cryogen-free instrument will use a commercial temperature controller to read the cryo-refrigerator’s first and second stage thermometers and provide feedback to a Proportional Integral Derivative (PID) loop regulating third party hardware: i.e. heaters, valves and/or vacuum pumps. In addition to these basic components, more complex cryogenic systems will in general contain “user-specific” electronic control modules. For example, systems used by physicists and chemists to perform research in materials science might also include a “bi-polar” power supply to charge and discharge a superconducting magnet, sensitive measurement electronics to read out Superconducting Quantum Interference Devices (SQUIDS) and other low temperature electronics. Any of these additional control modules will then communicate to the PC via a IEEE-488 (GPIB), USB or the CANopen interface.



Very often the electronic modules comprising any such complex cryogenic instruments are controlled by an operator via an application running on a personal computer (PC). A popular software package for general measurement automation used by our customers is LabVIEW from National Instruments [6]. This paper describes how the Quantum Design SDK [7] (for the GA-1 cryo-refrigerator system bridges between these two software environments for controlling the user's experiment and the cooling power of the cryo-refrigerator system.

2. Experimental results

Figure 1 (left) is a photo of the test station set used to characterize and experiment with the GA-1 cryo-refrigerator (A) in this study. The Cryogenic Test Chamber (B) is equipped with a Cryo-refrigerator pressure sensor (C), electrical feed through and a vessel pump out port (D). A modified Temperature Control Module (E) and Power Supply (F) are used to read diagnostic thermometers mounted on the first and second stages of the cryo-refrigerator. An Oscilloscope (G) and Laptop PC (H) are used to monitor and record raw pressure and displacer data and automatically generate load maps and cool down curves. A variable speed HAC 900S air cooled compressor (I) drives the GA-1 cryo-refrigerator.

Figure 1 (right) illustrates a block diagram of a LabVIEW or similar .NET software installed on a computer connected to a temperature controller, via for example, a GPIB or a USB interface. This interface, for example, will report to the computer temperature readings of the cryo-refrigerator first stage and the temperature of a conduction cooled superconducting magnet linked to the second stage.

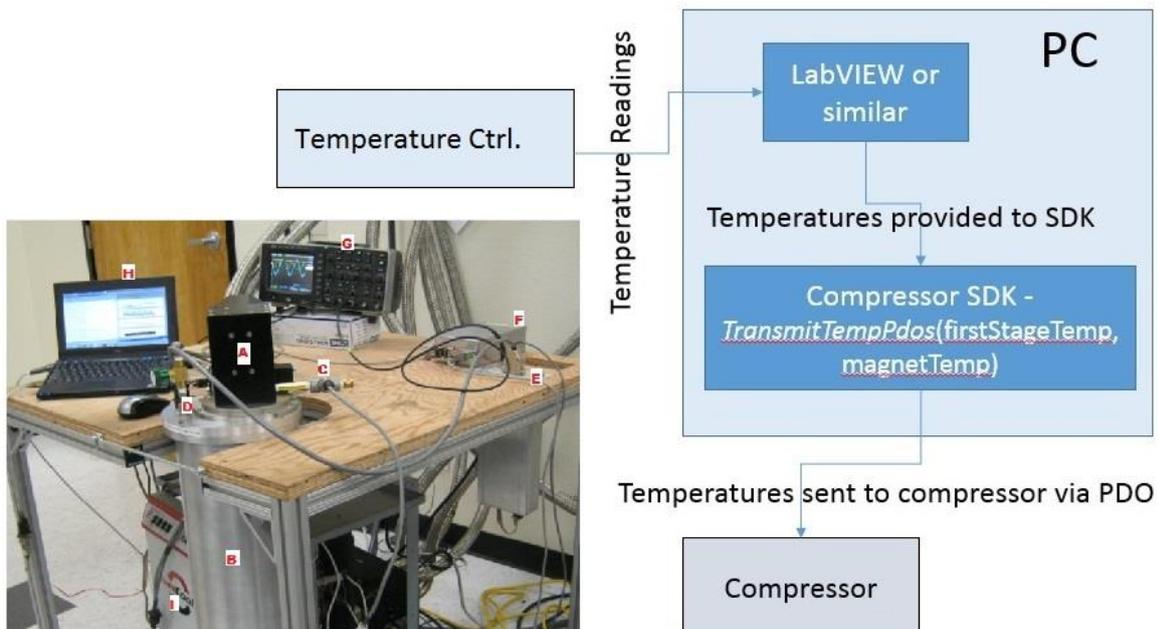


Figure 1. (Left) Photo of the cryo-refrigerator test station. (Right) Example block diagram of LabVIEW application running in conjunction with the Compressor SDK.

A LabVIEW, Visual Basic for Applications (VBA), C# or equivalent program interfaces with the Compressor SDK to control the cooling power delivered to the first and second stages of the GA-1 cryo-refrigerator system as required by the user requirements via CAN Process Data Object (PDO) commands. Provided below is an example of using the SDK to set custom compressor speeds, and make the compressor run remotely:

```
// Create a ControlComp object
Hac900s comp = new Hac900s();
// Enable remote control
```

```

comp.SetRemoteModeEnable(true);
// Set the speed mode to "CustomSpeed"
comp.SetRemoteModeType(CompressorMode.CustomSpeed);
// Set the head speed to 60 Hz
comp.SetCustomHeadSpeed(60);
// Set the compressor speed to 25 Hz
comp.SetCustomCompSpeed(25);
// Start the compressor running
comp.SetRunEnable(true);

```

An example of using C# and the SDK for acquiring temperatures of the first and second stage thermometers from a modified Quantum Design Temperature Control Module (Model CM-G) and transmitting them to the helium compressor using the TransmitTempPdos Application Programming Interface (API) as set point variables follows below:

```

private CanBus canCmd;
canCmd = new CanBus();

private void startTemperatureControl()
{
    pdoThread = new Thread(PDOUpdate); // create thread
    pdoThread.Start();
}

private void PDOUpdate()
{
    Byte TCM = 3;
    while (true)
    {
        float stage1T = 0, stage2T = 0;
        System.Threading.Thread.Sleep(500); // update every 0.5sec
        canCmd.ReadSdo<float>(TCM, 0x6121, 2, ref stage1T);
        canCmd.ReadSdo<float>(TCM, 0x6122, 2, ref stage2T);
        if (stage1T > 0 && stage2T > 0)
            easyComp.TransmitTempPdos(stage1T, stage2T);
    }
}

```

Other physical quantities of interest to the end user such as: the sample temperature, applied magnetic field strength and helium flow can alternatively be read from sensors and transmitted to the PC. These dynamic variables can then be used in a feedback loop to regulate the cooling power delivered from the cryo-refrigerator's first and second stage. Furthermore, a "quiet-mode" can be implemented to minimize the vibrations transmitted by the cryo-refrigerator displacers and valve movement to the experiment when the temperature is stable.

In addition to user specific physical measurements, a variety of cryo-refrigerator parameters can be monitored using this interface to ensure proper operation of the system. This setup is especially useful in mission critical experiments where an accidental warm-up of the system should be avoided. Some important parameters that can be monitored are: Supply and Return pressures, Compressor and Cold Head speeds, oil and helium temperatures at various locations of the compressor circuit, oil level and oil flow. The microcontroller on the compressor can use these parameters to adjust the operating conditions to be as efficient as possible. In practice, a running log of these "vital-signs" of the compressor can be maintained and monitored for diagnostic and trouble-shooting purposes. For instance, the Oil Flow Ratio (OFR) defined as:

$$OFR = (Cool\ Oil\ T - Cool\ He\ T) / (Capsule\ T - Cool\ He\ T) \quad (1)$$

Where *Cool Oil T*, *Cool He* and *Capsule T* are the temperatures of the oil, helium gas and capsule in the compressor gives a good indication of how well the oil is flowing and cooled in the helium compressor. For instance, if OFR=1, then the oil is not being cooled properly in the oil heat exchanger since *Cool Oil T* is equal to *Capsule T*. At the other extreme, OFR = 0, the *Cool Oil T* is equal to the *Cool He T*, which is an indication of no oil flow, or an inadequate oil charge that might result from aging of the Scroll Capsule or an oil leak.

2.1 Example of Graphical User Interface (GUI)

To demonstrate a thorough example of how to use the SDK Dynamic Link Library (DLL) we have developed a generic GUI that allows for the PC based control of the compressor functions. Figure 2, shows the “Select” tab of the QD Compressor Control Center. This tab contains basic and advanced commands which allow the user to access several compressor modes which include: Factory pre-set cooling power levels (High speed, Normal speed, Low speed), Custom speed, and PID Mode. On the right hand-side, real-time values of experimental and diagnostic variables read from sensors in the system are displayed for the user monitoring.

2.1.1 Factory Pre-set Power Levels

LowSpeed, *NormalSpeed*, and *HighSpeed* correspond to three factory “pre-set” power levels available in the compressor. The use of these modes is the simplest way to begin controlling the cooling power of the compressor and cold head in a custom application. The input frequencies at which the compressor capsule and cold head motor typically run and internal operating parameters in these modes are described in Table 1.

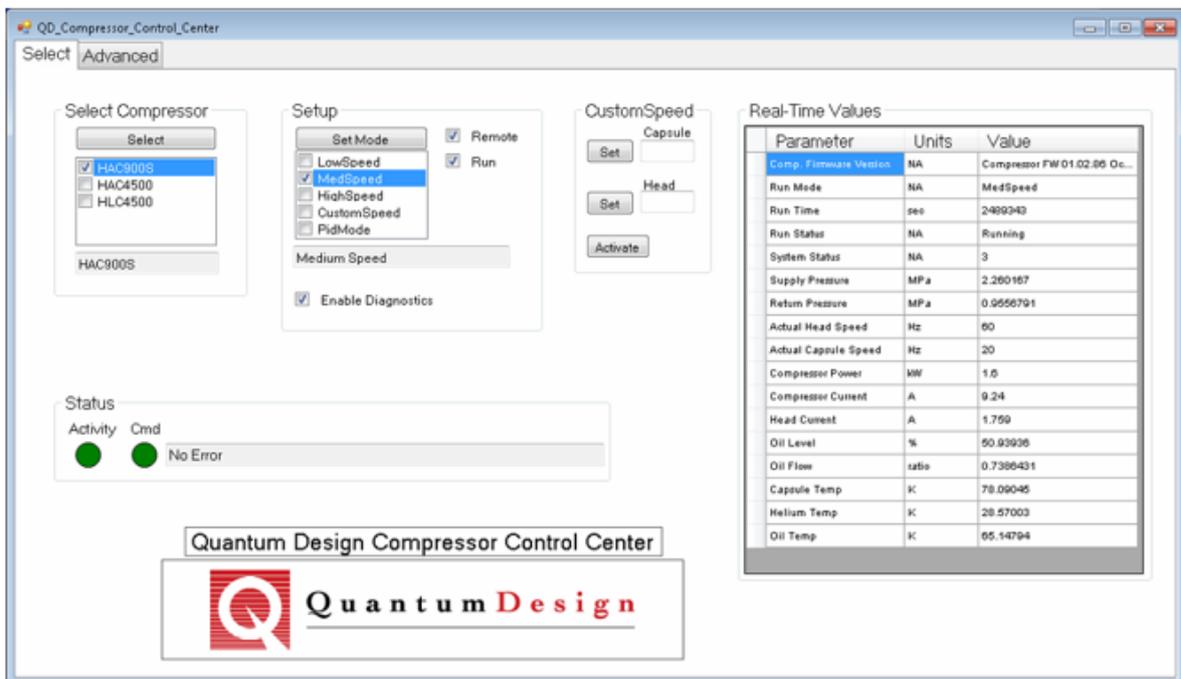
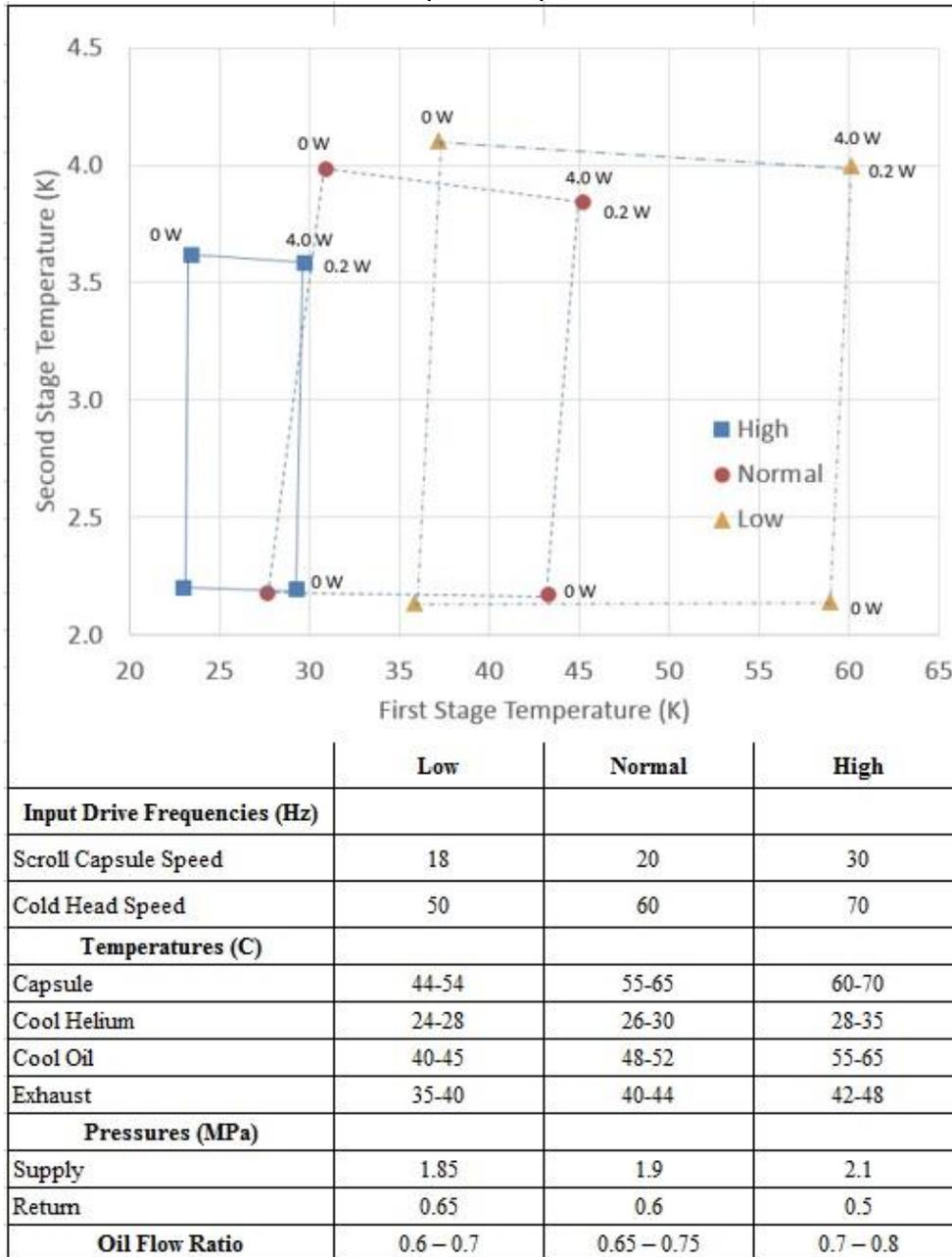


Figure 2. “Select” tab of the QD Compressor Control Center.

Table 1. Simplest Compressor Modes

2.1.2 Custom Speed Mode

With the *CustomSpeed*, the user specifies a cold head motor speed and compressor speed, in terms of input frequency, and in units of Hz. In general, using faster speeds yields faster cooling up to a certain point in which frictional losses become relevant causing a loss of cooling power.

2.1.3 PID Mode

With *PIDMode*, the firmware controls both the cold head and compressor capsule speeds using a Proportional-Integral-Derivative (PID) control algorithm. The process variables used in the PID control are the cold head first-stage and second-stage temperatures. In general, this mode depends upon first and second stage target temperature set-points, easily set by an object command, as well as PDOs reporting the current first-stage and second stage temperatures. By default, the set-point temperatures

are 40 K for the first stage and 4 K for the second stage. The software controls temperature by ramping the compressor and head speed up or down.

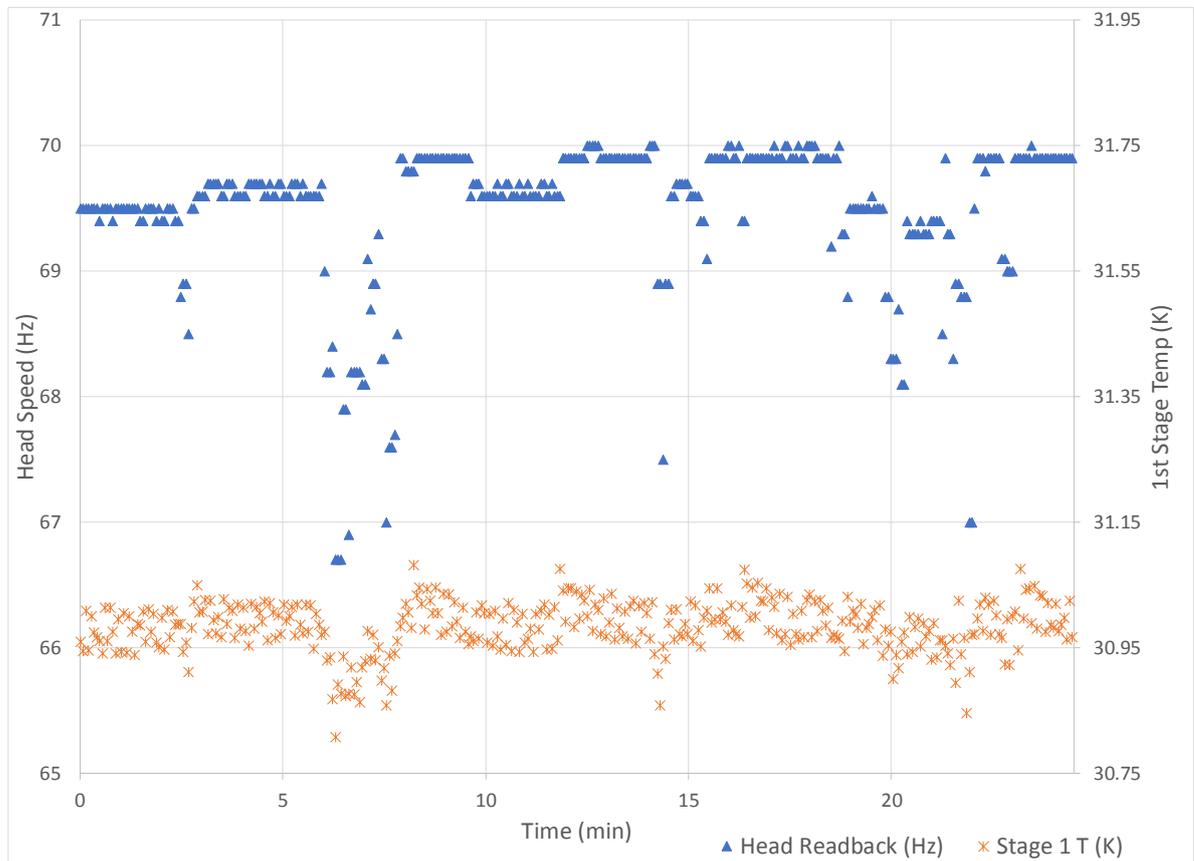


Figure 3. System in PID mode, varying head and compressor speeds to maintain a first stage target temperature of 30.95 K.

2.1.4 Real Time Data Monitoring and Logging

In addition to speed control and temperature read back previously mentioned the SDK and GUI allow real time monitoring of several other system parameters such as: draw, shuttle valve travel and the phase delay the shuttle valve and cold head displacers. This data can be saved to a .csv formatted data file with a user selectable logging interval.

2.1.5 Warming Mode

The dynamic control of valve timing using a microprocessor allows for the GA-1 cryo-refrigerator to provide cooling and warming of the first and second stages in a “on demand” basis [1]. This form of temperature control is attained without the use of external heaters. In practice, this is accomplished by reversing the phase of the intake and exhaust functions during the refrigeration cycle.

This feature is particularly desirable if a fast temperature cycle of the cryo-refrigerator system is desired. This might be the case if a high temperature ($T > 77$ K) regeneration of a cryo-pump, linked to the first stage of the cryo-refrigerator, and subsequent cool down is required. Previously, this operation might require for the entire cryo-refrigerator system to be shut down for a several hours to allow it to warm up. In our system, the cycle can be achieved in 30-40 minutes. Using the GA-1 cryo-refrigerator, cooling and heating of the first and second stages can be simply achieved by varying the

Phase Delay (PD) of the intake and exhaust functions during the refrigeration cycle. Figure 4 (upper) shows a cooling curve attained with no heat load and a PD = 55°. It takes 70 minutes for the first stage to reach 40 K and about 60 minutes for the second stage to achieve temperatures slightly below 2.5 K. The figure inset shows oscilloscope traces of the “clock signal” given by the reciprocating motion of the displacers and the gas exhaust and intake profile followed by the spool valve. On the other hand, figure 4 (lower) shows that from base temperature and no heat load, with a PD = 235° it takes about 35 minutes for the first stage to reach 300 K. The data shows warming can be achieved by solely changing the phase delay at a fixed speed.

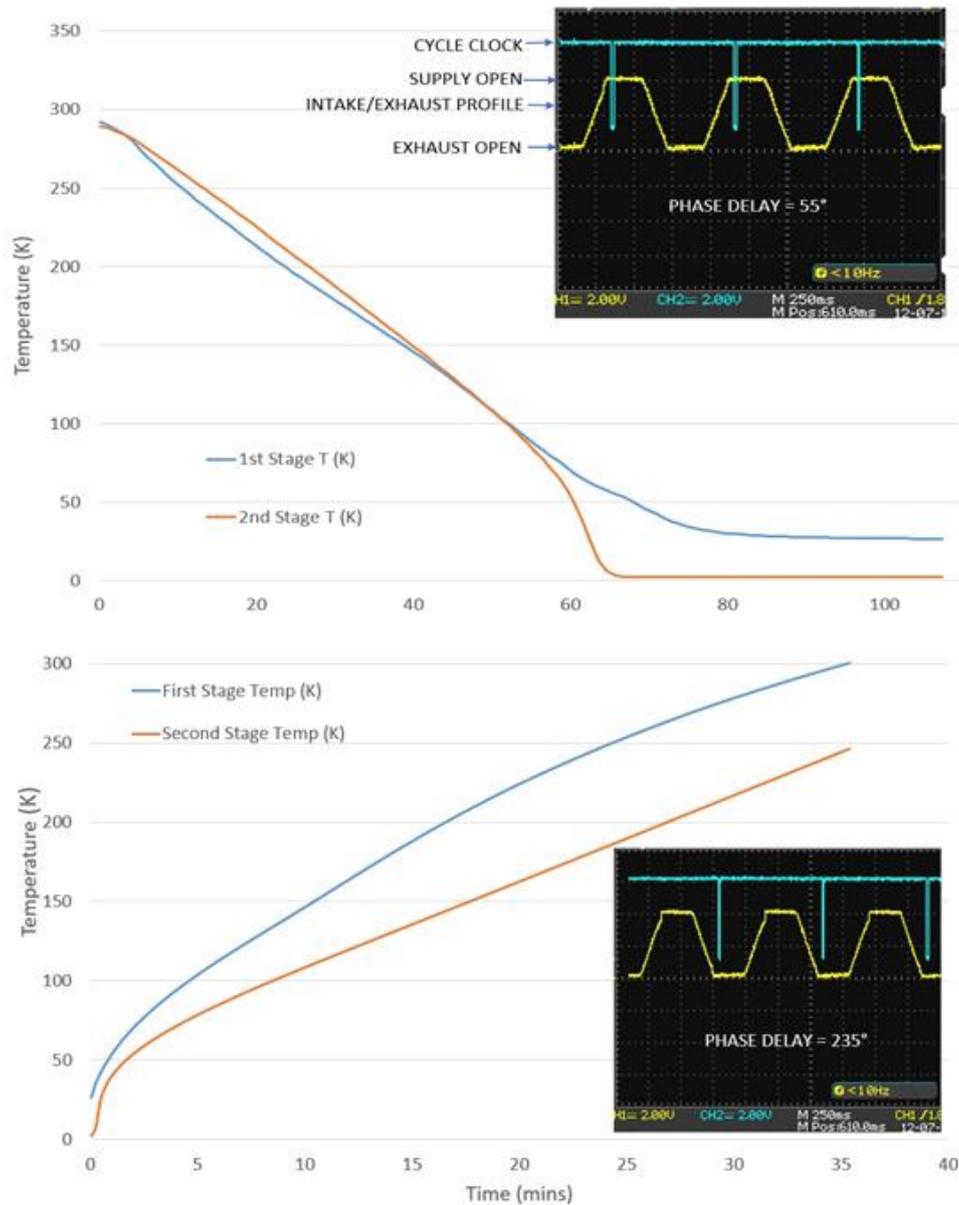


Figure 4. Coll Down and Warm up curves.

3. Conclusion

In this study, we have demonstrated a remote interface kit for integration of the GA-1 energy smart cryo-refrigerator with 3rd party hardware and software. Through speed and phase delay control, the cooling power delivered by the cold head can be optimized for the needs of various applications such as SQUIDS, low temperature amplifiers or cryopumps. Further development of the PID control algorithm will incorporate additional variables such as phase delay and valve travel distance to provide enhanced automated temperature control. Additionally, a “quiet mode” function will be added to the GUI for reduced cold head vibrations when operating at cold temperatures.

4. References

- [1] Snow D, Lawton J, Taff R, Diederichs J, Gardiner J, Simmonds M B, Hamilton J and Spagna S 2017 *Cryocoolers 19*, ICC Press, Boulder, CO
- [2] Sayles T, Martien D, Diederichs J, Sloan J, Bird D and Spagna S 2011 *Cryocoolers 16*, ICC Press, Boulder, CO
- [3] Chialvo C, Sayles T, Diederichs J, Spagna S, Gabal M., Sesé J and Rillo C 2014 *Cryocoolers 18*, ICC Press, Boulder, CO
- [4] Simmonds M B and Pfeiffer O 2002 *iCC Conference Proceedings*
- [5] Model VersaLab Physical Measurement System, manufactured by Quantum Design Inc.
- [6] *Quantum Design Application Note* 1070-210.
- [7] *Quantum Design Application Note* 1201-017