

Overview

Creating custom software for the mK2000 temperature controller is remarkably simple, thanks to the usage of SCPI (Standard Control for Programmable Instruments) command format, and the integration of serial port drivers into modern operating systems. To aid in this process, Instec has created a number of software development resources, including sample code and a command reference guide. These resources can be accessed using the links below. It is recommended to copy and paste these links directly into the search bar of your preferred web browser.

Command Reference guide

The full list of SCPI commands supported by the mK2000 temperature controller are described in the command reference document. The command reference guide also contains information on the SCPI format, specifically how to parameter syntax and the command path structure.

Link : https://www.instec.com/programs/sample-codes/MK2000B_SCPI_cmd_v4.00_Basic.pdf

mK2000 Temperature Controller Theory of Operation

The mK2000 temperature controller is used to set and maintain the temperature of an Instec temperature-controlled device such as a thermal stage, plate, or chuck. To control the temperature, the mK2000 uses a PID algorithm that is calculated entirely on the controller side: all that must be done from a user perspective is to tell the controller what temperature is desired. To execute the PID algorithm, and to communicate with the user via the keypad/touchscreen or a Host application, the mK2000 temperature implements a Real Time Operating System (RTOS).

From a programming perspective, operation of the mK2000 temperature controller can be somewhat neatly categorized into 3 phases; Connection, Setup and Operation. The details of each phase can vary greatly depending on the application, but the Connection and Setup phases remain mostly the same.

1. Connection

During the connection phase, a line of communication is established between the mK2000 temperature controller (Device), and the personal computer (Host) executing the custom software. Typically, communication is done via a SCPI commands sent over a Virtual COM port, aka Virtual Serial Port.

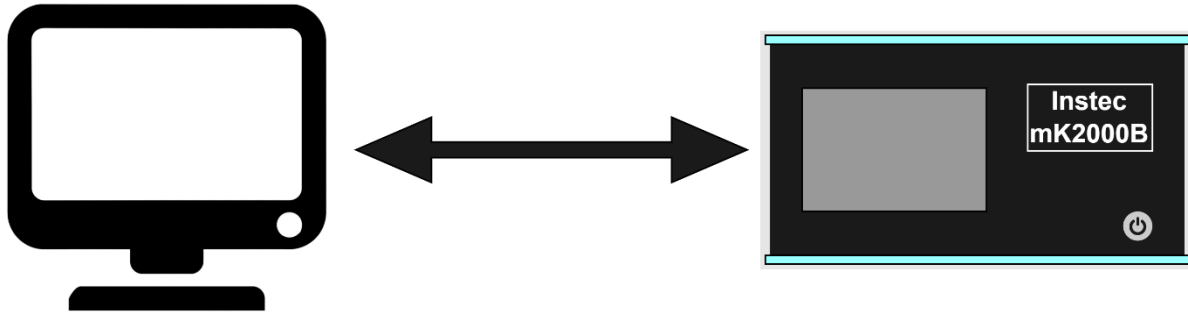
2. Setup

Setup for the mK2000 is preformed after communication between the Host and the Device. This setup typically includes retrieval of the key parameters from the device, including temperature range, device information, and PID tuning. These parameters are stored on the device, but can be read/ changed using a Host software.

3. Operation

Operation of the mK2000 includes sending operation commands, such as Hold at X temperature (TEMP:HOLD X), as well as requesting runtime information (TEMP:RTIN?). Runtime information requests are typically sent repetitively, i.e. Every 1 second, so that a constant stream of temperature data can be collected. Operation commands need only be sent once per desired operation. In a temperature profile, where the goal is to regulate the sample temperature according to an arbitrary temp vs time curve, a unique operation command should be sent at each inflection point. With customized software, logical operations can be conducted as well, including loop structures, if-statements, and logic structures. In general, regardless of the application, the operation phase is concluded with a "Stop-all" command, which returns the device back to the idle state before disconnecting the Device from the Host.

mK2000 Operation Example



Connection Phase - Example

Step Number	Host Application	USB Connection	Device RTOS
1.A	Host initiates connection (Method varies based on software type)	→	Initiates VCP connection
Etc...			

Setup Phase - Example

Step Number	Host Application	USB Connection	Device RTOS
1.A	Sends string “*IDN ?”	→	Processes SCPI Command
1.B	Receives string “<Manufacturer>, <device>, <SN>, <FW Version>”	←	Sends ID Information
2.A	Sends string to request other mK2000 information, ie. “Temp:RANGe?”	→	Processes SCPI Command
2.B	Receives String “<Upper limit>, <Lower Limit>”	←	Sends Temperature range information
Etc...			

Operation Phase - Example

Step Number	Host Application	USB Connection	Device RTOS
1	Host software sends “TEMP:HOLD X”	→	Processes SCPI Command, temperature control begins
2.A	Host sends “TEMP:RTIN?”	→	Processes SCPI Command
2.B	Receives and parses runtime info string	←	Sends Runtime Information
3.A	1s elapses, Host sends “TEMP:RTIN?”	→	Processes SCPI Command
3.B	Receives and parses runtime info string	←	Sends Temperature range information
4.A	1s elapses, Host sends “TEMP:RTIN?”	→	Processes SCPI Command
4.B	Receives and parses runtime info string	←	Sends Temperature range information
Time elapses, temperature stabilizes			
5	User makes measurement, Host sends “TEMP:STOP”	→	Processes SCPI Command, stops temperature control, returns to Idle Mode
6	Connection is terminated		

Sample Code

Python

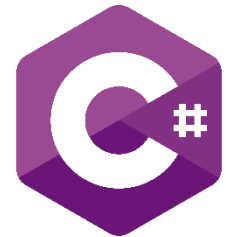
The Python sample code for mK2000B includes a full Python library for interacting with the mK2000B, and a number of examples demonstrating the full range of capabilities of the temperature controller. The Python SDK is recommended for mK2000B units, but will work with reduced capability for most mK2000VCP models. Detailed setup and operation instructions are included on the Instec GitHub page, which is linked below.



Link: <https://github.com/instecinc/pyinstec>

C # (C-Sharp)

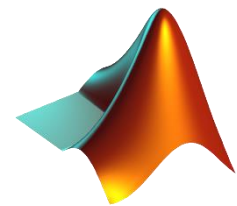
Our C # sample code implements a simple GUI based on the .WPF framework. The GUI demonstrates COM port connection and some of the most important SCPI commands, essentially comprising a simple COM port debugging app with a few built-in mK2000 features. The demo is built from Microsoft Visual Studio, which available for free online.



Link: <http://www.instec.com/programs/sample-codes/MK2000-VCP-Csharp.zip>

MATLAB

Our MATLAB sample code is a simple example meant to demonstrate how to connect the mK2000 and send basic commands via the built-in "Serial" library. Just copy and paste the text from the Word doc into a MATLAB script, and save a local copy onto your machine before executing. To repeatedly send the read temperature commands, is it recommended to use a repeating structure such as a while loop or a timer.



Link: www.instec.com/programs/sample-codes/mk2000VCP_matlab_example_code.docx

LabVIEW

Our sample code for LabVIEW implements a full GUI solution very similar to our InstecApp desktop application. The sample code requires an active version of LabVIEW 2015 or later to execute. The application as a whole can be embedded into a custom application, or the Sub VI's can be reverse engineered to be used with a custom LabView application.



Link: [www.instec.com/programs/sample-codes/Labview for MK2000 and MK2000B\(SCPI\).rar](http://www.instec.com/programs/sample-codes/Labview%20for%20MK2000%20and%20MK2000B(SCPI).rar)

Note: if needed, the driver installation guide for the mK2000 can be downloaded here:
[www.instec.com/pdf/How-to/InstecApp WinTemp Driver installation Guide V1.3.pdf](http://www.instec.com/pdf/How-to/InstecApp_WinTemp_Driver%20installation%20Guide_V1.3.pdf)