Pico Commander

Proposal Document

Author: Albert Jansen

Prepared for: Randy Gallant

Course Code: ELIC350

Date: February 10, 2016

Table of Contents

Proposal

User Manual

Introduction

The Pico Commander

The Pico Commander is an all in one controller solution, giving user the ability to control real world systems from their desktop. Users will be able to easily create and upload programs to the Pico Commander using the Pico Commander desktop application. These programs will control 8 digital outputs, read 8 digital inputs, and do a variety of on-chip processing tasks. With intuitive program creation and high output drive capability, the Pico Commander will bring simple control solutions to the open source community.

Project Proposal Overview

This document contains a technical and project management overview of the Pico Commander project. The technical side of the proposal analyzes the major technologies in the Pico Commander by breaking down a high level block diagram of operation. As various technologies are introduced, the reason for their selection and the impact they will have on the final product is discussed.

After analysing the high level block diagram and explaining the technical design of the project, the focus shifts to project management. Major technical specifications and branding details are given, as well as timelines and budgets. The conclusion of the document reviews the project philosophy and project planning. Following the conclusion is a brief introduction to the User Guide.

Block Diagram and Project Technologies

The following figure visually describes the high level systems of the Pico Commander. The project is divided into two parts, linked by a USB communications channel. The first component is a desktop application (yellow). This application allows the user to interact with the Pico Commander by creating, uploading, and monitoring programs. The Pico Commander controls 8 inputs and 8 outputs (blue), based on the commands interpreted from the uploaded user programs. These inputs and outputs have a buffer stage to provide strong over-voltage protection and high output power. The Pico Commander is connected to the computer via a USB link (grey). The other grey link is the SPI breakout, which provides support for modular expansion boards.

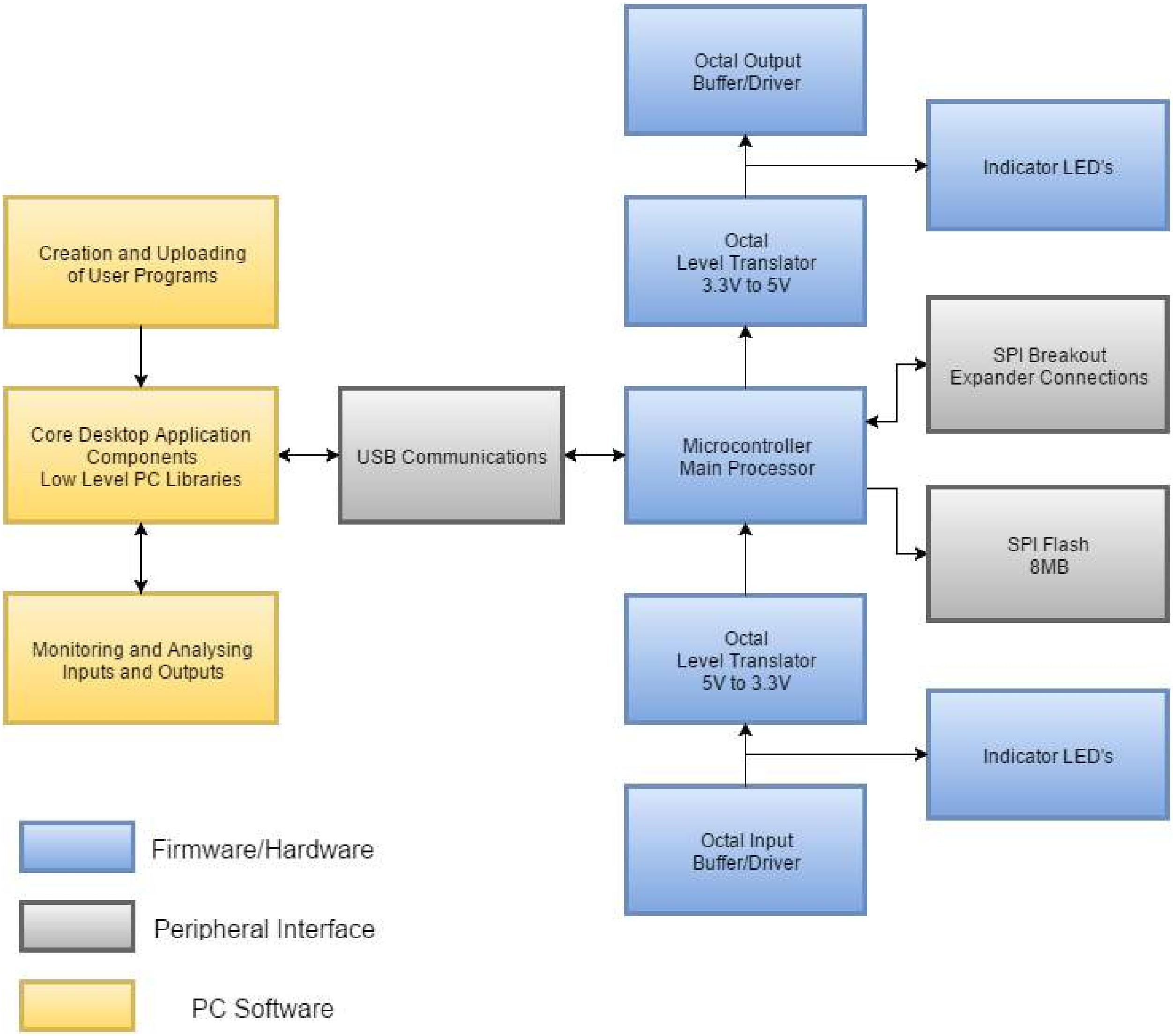

Octal Output
Buffer/Driver
Indicator LED's
Creation and Uploading of User Programs
Octal Level Translator 3.3V to 5V
SPI Breakout Expander Connections
Core Desktop Application Components Low Level PC Libraries
USB Communications
Microcontroller Main Processor
SPI Flash 8MB
Monitoring and Analysing Inputs and Outputs
Octal Level Translator 5V to 3.3V
Indicator LED's
Octal Input Buffer/Driver
Firmware/Hardware
Peripheral Interface
PC Software

Figure 1: High Level Block Diagram of Pico Commander

PC Communications and Program Creation Software

This is the core component of the desktop application. The software will be written in Python, implementing a variety of libraries to enable USB/Serial port communication and GUI development. This first block contains the lowest level of software, and will encapsulate all of the functions the user will not interact with directly, such as: driver installation, USB communication, and the core GUI functionality.

Python's extensive library support is one of the reasons that it was selected. PyQt4 and PySerial are the libraries being used for GUI and USB development respectively. By using libraries to interface with the lower levels of the PC, the Pico Commander software development can focus on creating a user interface and higher level communications functions.

User Program Creation and Uploading

Programs will be created by a user in the GUI of the desktop application provided with the Pico Commander. Two program creation options will be available: Drag and Drop, and Hex Opcode. Drag and Drop program creation is further subdivided into two categories: Block Diagram, and Ladder Logic. Traditional block diagram drag and drop allows new users with minimal programming experience to get started quickly, while creating a program using Ladder Logic allows users who are experienced in control systems to port old programs from other platforms comfortably.

Hex Opcode program creation will allow the user to directly enter commands and addresses in hexadecimal format. More detail on the Hex Opcode format, including a table of commands and memory will be given in the Technical Reference Manual. User programs will "compile" to a .upl file, which is a compressed version of the Hex Opcode format. The .upl files contain the data that will be uploaded the Pico Commander.

Monitoring and Analyzing Inputs and Outputs

There are two methods of monitoring inputs: Automatic and Requested. Both options can monitor the inputs and outputs of the Pico Commander and either write the results to a local file, or send the results to the PC for saving, live monitoring, and more.

Analysis can also be done on the inputs. If the Pico Commander is simply being used as a data acquisition system, the PC software will provide functions for the user to extract meaningful results from their data, and display those results accordingly.

Microcontroller and Main Processor

At the heart of the Pico Commander is a Texas Instruments C2000 TMS320F28069F Microcontroller. This microcontroller provides the data processing, I/O monitoring, and communications functions requested by the user in their program.

The microcontroller was selected for a variety of the features that it contains. First, it has an on-chip USB peripheral and firmware stack. This eliminates the need for an external USB chip, increases the reliability of the USB functionality, and reduces USB firmware development complexity. Secondly, it has a co-processor which works in parallel with the main processor. This allows the main processor to focus on communications tasks, while the co-processor handles input interrupts and some math processing. Finally, it is designed for control applications, and many of the resources that are provided with the chip are directed towards this field.

Input and Output Stages

These stages are denoted in the block diagram by "Octal Level Translator", "Octal Buffer", and "Indicator LED's". These multi-stage input and output sections provide higher current and voltage capabilities than those native to the microcontroller. The input stage is designed to protect the microcontroller, and the output stage is designed to offer as much power as possible, while maintaining a small form factor and not increasing the analog complexity. The indicator LED's can be used as a quick reference to monitor the inputs and outputs.

USB Communication

A USB 2.0 connection will be used to monitor the inputs, upload programs, and interact with the Pico Commander. There will be a driver provided with the desktop application that will allow the computer to recognize the Pico Commander. The aforementioned methods of monitoring inputs (Automatic and Requested) will be implemented based on the availability of data transfer functions in the USB firmware.

SPI Breakout Headers

The future development of the Pico Commander system will take place via expansion boards that can be connected via a SPI peripheral that is broken out. Details on how to design boards to fit with this connector will be outlined in the Technical Reference.

Project Technical and Branding Specifications

Technical Specifications

Specification	Max.	Units
DC Jack Input Voltage	5.25	V
Octal Input Voltage	5.5	V
Octal Output Voltage	5.5	V
Octal Output Source Current	80	mA
Octal Output Sink Current	180	mA
Octal I/O Toggle Frequency	20	MHz
USB Speed	12	Mbps
Maximum Program Size	4	MB

From the table above, one can see that the Pico Commander is a 5V system, and is capable of providing and reading input frequencies of up to 20MHz. The user created programs must compile to under 4MB, and the output stage contains a high current buffer that allows the Pico Commander to provide higher than standard levels of output power. For more information, refer to the User Guide and Technical Reference.

Renderings and Branding

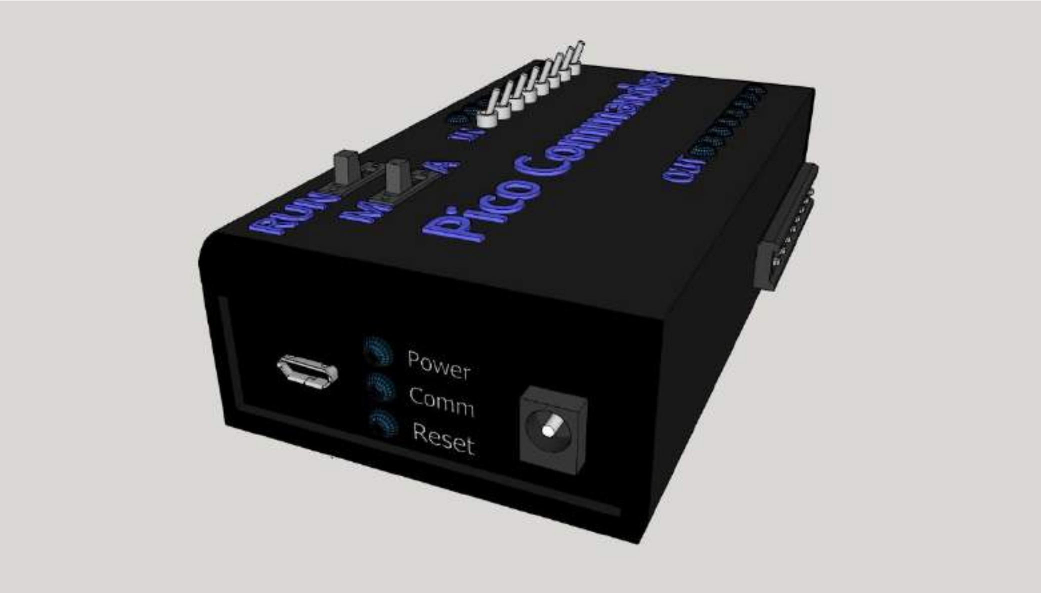


Figure 2: Front panel rendering, including USB port, DC Jack, and LED Indicators

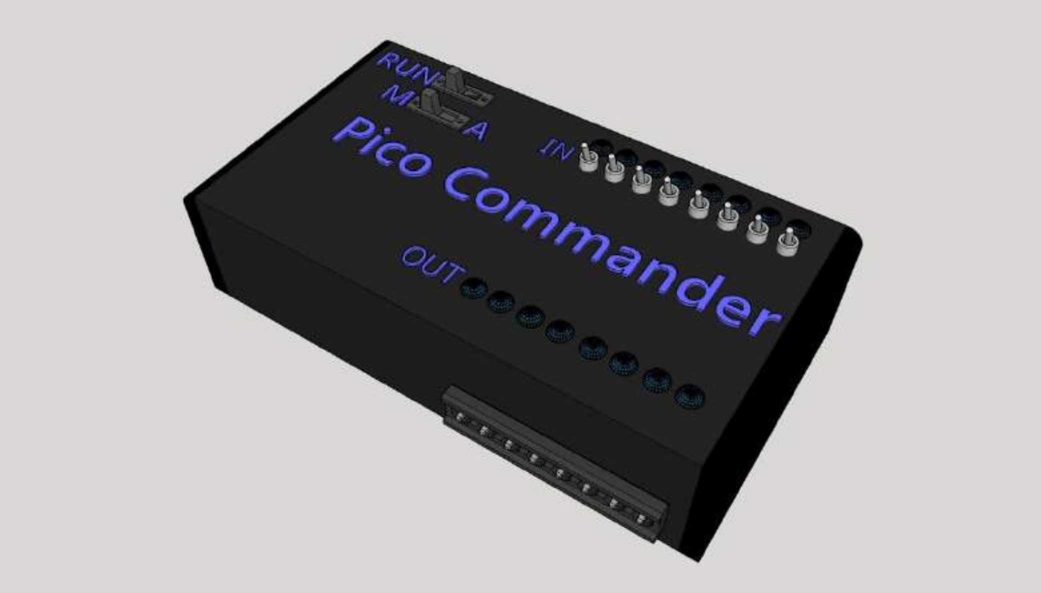
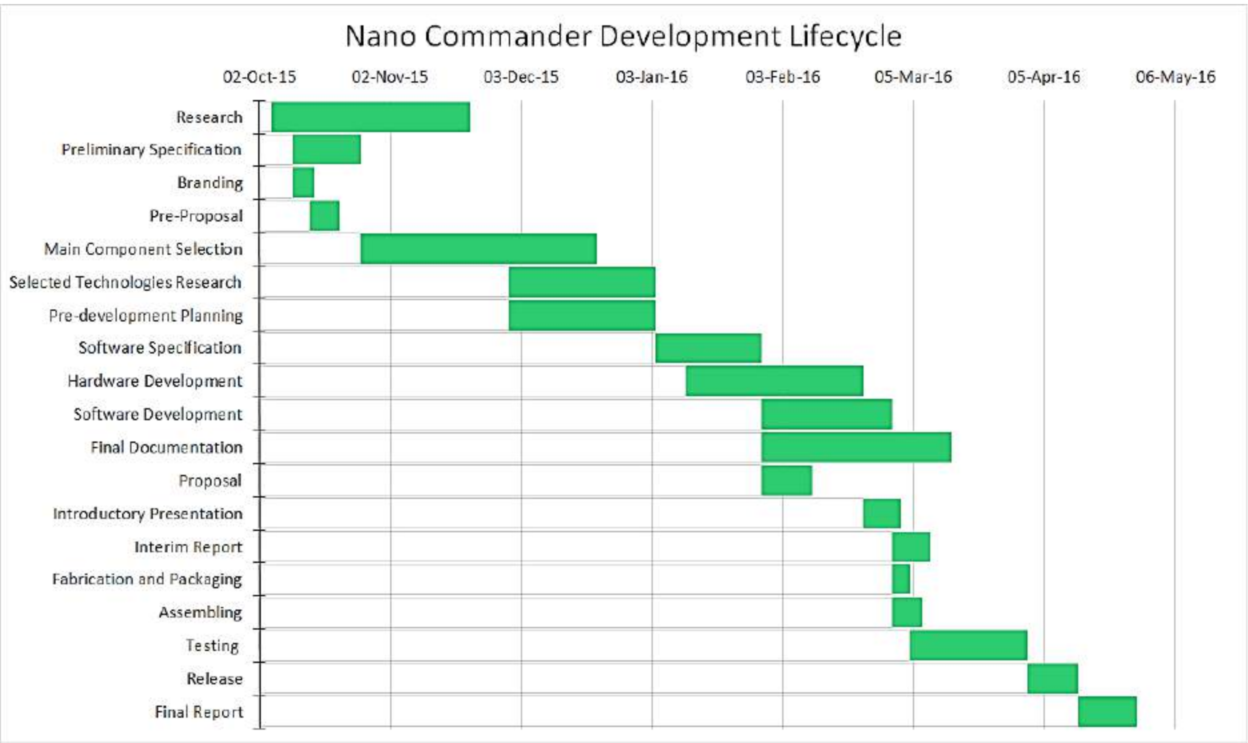


Figure 3: Rear panel rendering, displaying toggle switches and I/O Ports

Above are two concept designs for the Pico Commander. The user will have the option to select if they want to use the toggle switches for input, or if they want the input connector to be used for the inputs. The indicator LED’s are shown, as is the front panel, containing a DC Jack, USB female connector, and several more indicator LED’s.

Project Management

Gantt Chart



Future Development

As outlined in the block diagram description, room will be provided to expand upon the project going forward in the form of the SPI expansion headers. These headers will allow easy, modular expansion through small firmware changes.

Beyond this simple form of extra development, future development also includes the addition of wireless communications on the board. These communications would be used for all of the same functions fulfilled by the USB communication link. The final PCB design will have room to add wireless functionality in the future without increasing fabrication cost.

Budget

Cost Type	Part Name	Quantity	Cost
Incident	Wall Wart Power Supply	1	13.58
	F28069 LaunchPad Development Kit	1	38.12
	XDS100v2 JTAG Debug Probe	1	79
Active Components	C2000 Piccolo Microcontroller	1	28.83
	Level Translator	2	3.92
	I/O Buffer/Driver	2	20.98
	1-Bit Level Translator/Buffer	1	1.16
	8-Bit Level Translator/Buffer	1	2.86
	Quad SPDT Analog Switch	2	2.4
	2:4 Active Low Decoder	1	0.85
	8MBit SPI Flash Memory	1	1.52
	LDO Voltage Regulator	1	2.27
Passive Components	Blue Indicator LED	20	8.2
	Other Passive Components	50	10
Fabrication and Assembly	Acrylic	1	10
	PCB Fabrication	5	80
	Stencil	1	20
	Solder Materials	1	20
TOTAL			343.69

Conclusion

The Pico Commander is a project that focuses on bringing high quality software, hardware, and firmware to the open source community. The project leaves plenty of room for future expansion, and the proposed timeline covers all of the essential points that will define the product. Moving forward, hardware and software development are the next stages currently underway.

Several components have already been ordered, and the fabrication costs will be kept to a minimum due to the availability of rapid development in the prototyping lab. assessment will be done on February 29th to re-evaluate the scope of the project, and what can be reasonably accomplished before the end of the semester.

User Guide

Software Instructions and Guide

Acquiring the Software

The software can be download from the following link: [INSERT LINK]. This is the .exe version of the software. If you execute this program, it will run the Pico Commander desktop application. The driver can be installed from within the desktop application.

Alternatively, you can run the source code yourself. The repository is found here: [INSERT LINK], and is free to use and distributed under the LGPL. You must at least Python 3.4 installed, and have PyQt4 and PySerial correctly configured and installed. Additionally, you will need to manually install the driver, which can be found here: [INSERT LINK].

Before Connecting the Pico Commander

Prior to connecting the Pico Commander, ensure that the software is working correctly. When you double click the PicoCommander.exe file, the GUI should open. If it does not, please ensure that you have downloaded the correct version and that you have permission to run files on your computer.

Next, install the driver that you downloaded in the first step. If you have the Pico Commander desktop application, simply select "Help > Drivers > Install NC Driver". Once you have selected this option, follow the on screen prompts that will assist you with installing the driver.

If you have downloaded the driver manually, simply right click on the .INF file and click "Install".

Developing Programs

Drag and Drop – Block Diagram

1. Begin a new project by going to File > New Project
2. In the project creation wizard, name your project, and select the directory in which you wish to save the project. Then click "Next".

3. From the “Creation Type” drop down, select “Block Diagram”. Leave the rest of the options untouched. Click “Next”.
4. Leave the next set of options untouched. Click “Next”.
5. Click “Finish”.

At this point, you should have a new program created. To add a block, select the “Add Block” button from the toolbar, and browse the library for the block that you wish to use. Click add, and place the block in the program creation workspace. You will need to use the “Add Wire” tool to connect blocks together.

Drag and Drop – Ladder Logic

1. Begin a new project by going to File > New Project
2. In the project creation wizard, name your project, and select the directory in which you wish to save the project. Then click “Next”.
3. From the “Creation Type” drop down, select “Ladder Logic”. Leave the rest of the options untouched. Click “Next”.
4. Leave the next set of options untouched. Click “Next”.
5. Click “Finish”.

At this point there you should have a new project open in the application. To begin your program creation, select the “Add Component” button and browse the library for the component that you wish to use. Once you have found the component, select “Add” and place the component where it will be connected. You will need to use the “Add Wire” tool to connect components on the ladder.

Hex Opcode

1. Begin a new project by going to File > New Project
2. In the project creation wizard, name your project, and select the directory in which you wish to save the project. Then click “Next”.
3. From the “Creation Type” drop down, select “Hex Opcode”. Leave the rest of the options untouched. Click “Next”.
4. Leave the next set of options untouched. Click “Next”.

5. Click "Finish

There will now be a text editor available to you. The editor itself has minimal code highlighting. Refer to the technical reference manual and example projects for more information on specific commands and how to format the Hex Opcode file.

Monitoring the Pico Commander

While the Pico Commander is operating and a connection has been established, there are several ways to view data on the inputs or outputs.

Live View

This view will draw a representation of the Pico Commander, and show you the current state of it's inputs and outputs. This view is useful for troubleshooting the Pico Commander, and testing basic programs. To view the Pico Commander inputs and outputs using Live View, select "Live View" from the monitoring toolbar.

Streaming View

If data is being placed on the inputs, you are able to have that data streamed to a file and saved on your computer. Select "Streaming View" from the monitoring toolbar to view streaming data from the Pico Commander. Alternatively, you can set the source of the data to be a text file, and view the streaming data in a static timing diagram view.

Requesting a Read

To request that the current state of the inputs and outputs be read, select the "Read Now" button in the monitoring toolbar. A text based format will appear that will give the current state of the inputs and outputs.

File Formats and Manual Configuration

Pico Commander uses a variety of non-standard file formats to keep programs organized. They are listed below along with their function.

.cfg – Project Configuration File

The .cfg files are text based files that contain the core information vital to a project, which includes:

- Project Name
- Relevant File Paths
- Board and Board Peripheral Hardware Specifications
- Project Build Status

This file type is editable by any text based editor, and more information on their structure can be found in the technical reference manual.

.brp – Board Properties File

These files contain information about the hardware being used in the project. This information is used during compiling and simulation to ensure that hardware integration of the Pico Commander will be as seamless as possible, and that the Pico Commander is capable of providing the proper hardware functionality requested.

.pro – Block Diagram Editor Save File

These files are used for storing the format of a program created in drag and drop block diagram mode. It saves the positions and connections made by the user. When the user wants to compile, the program uses the information in the .pro file to compile a .upl file.

.lbr – Contains Info About Programming Libraries

These files contain the relevant information about libraries that users can create projects with. They must be imported before a library of functions can be used. This is done automatically by the program (unless the user has selected Hex Opcode format program creation).

.upl – Pico Commander Hex Instruction File

These files contain the data that will is uploaded to the Pico Commander. They are Hex Opcode, assembly style format. Further details and in depth analysis can be found in the technical reference manual.

Compiling and Uploading

Before compiling, be sure to save your project.

Compiling

1. Save your project
2. Click the "Compile" button in the Pico Commander toolbar

All projects, regardless of program creation selected must be compiled before they can be uploaded. The compiling process creates a .upl file, which contains the hex data that will be uploaded to the Pico Commander.

Uploading

1. Ensure that you have compiled your project
2. Click the "Upload" button in the Pico Commander toolbar.
3. DO NOT DISCONNECT the Pico Commander until upload is complete.

Once the upload is complete, the Pico Commander will begin to run your program. To restart the program, press the "Restart" button on the Pico Commander.

Hardware Instructions and Guide

Overview of the Pico Commander Hardware

The Pico Commander has 8 inputs and 8 outputs, each capable of sinking 180mA of current and sourcing 80mA of current. The inputs are capable of reading a maximum toggle frequency of 20MHz, and the processor runs at 90MHz. SPI Communications run at 22.5MHz, however this can be adjusted depending on the peripherals that are connected.

On the front panel there is one USB port and one DC Jack. The USB port is used for connecting a PC, while the DC jack is used for connecting power. If the device is being powered via the USB port, it automatically enters low power mode, and the sinking sourcing capabilities of the Pico Commander are greatly reduced. For this reason, it is recommended that while the Pico Commander is operating it is connected to a 5V DC connection via the DC jack capable of supplying 2A of current.

To begin running a program, one must slide the program switch to the "RUN" position. This switch is implemented so that the Pico Commander does not draw too much power from a USB port when it is connected.

To control the inputs via the toggle switches, simply slide the input selection switch to "MANUAL".

Connecting to a PC

Before connecting the Pico Commander to the PC, ensure that you have successfully set up the software and installed the driver. If the software is correctly set up, connect the Pico Commander to the USB port on your PC. The power light will come on, and then the lights on the front panel will flash 3 times. If this happens, the Pico Commander is functioning correctly.

Connecting Inputs and Outputs

When interfacing the Pico Commander in a real world application, care must be taken to not overload the system. Each output pin can source 80mA, for a total of 640mA being sourced by the Pico Commander.

The inputs should be connected in such a way that the input current is at a minimum. While the inputs can sink 180mA, all the inputs should not attempt to sink this level current simultaneously. The maximum total current sink is limited to the same value as the output total current source: 640mA.

Safety and Maximum Ratings

Maximum Ratings

Specification	Max.	Units
DC Jack Input Voltage	5.25	V
I/O Voltage Range	0 - 5.5	V
Single Pin Source Current	80	mA
Single Pin Sink Current	180	mA
Maximum Total Output Current (DC Jack Power)	640	mA
Maximum Total Output Current (USB Power)	300	mA
Octal I/O Toggle Frequency	20	MHz
USB Speed	12	Mbps
Maximum Program Size	4	MB

Safety Warnings

Caution must be taken during the operation of the Pico Commander. The currents that can be supplied are high and potentially dangerous. Verify all circuits before running program, and ensure that power jacks and connections are secure. Complete routine maintenance checks on all relevant electrical systems.

References

- Data Sheet, Texas Instruments, "TMS320F2806X Piccolo Microcontrollers," TMS320F28069 datasheet, Nov. 2010 [Revised July 2014].
- Data Sheet, Texas Instruments, "25-Ω Octal Buffer/Driver With 3-State Outputs" SN64BCT25244 Datasheet, Dec. 1992 [Revised January 1994].
- Technical Reference, Texas Instruments, "TMS320x2806x Piccolo Technical Reference Manual," TMS320F28069 Reference Manual, Jan. 2011 [Revised March 2014].
- Reference Design, Texas Instruments, "Factory Automation and Control System Solutions", 2012. Available from: http://www.ti.com/lsds/ti/apps/automation/overview.page
- Reference and Technical Solutions, Maxim Integrated, "Programmable Logic Controllers", 2013. Available from: https://www.maximintegrated.com/en/solutions/industrial/control-and-automation/plc.html/tb_tab1
- Pit Documentation, Python Wiki, "About PyQt", June 2015. Available from: https://wiki.python.org/moin/PyQt
- PySerial Documentation, Chris Liechti, "Welcome to pySerial's documentation", 2015. Available from: https://pythonhosted.org/pyserial/
- Beyond Logic, Craig Peacock, "USB in a Nutshell", September 2010. Available from: http://www.beyondlogic.org/usbnutshell/usb1.shtml
- Taylor Singletary, "Writing Great Documentation", January 15, 2016. Available from: https://medium.com/life-tips/writing-great-documentation-44d90367115a#.sz21wcryi