**Overview**

The purpose of the Command Tester board is to allow a user to send commands to various devices for testing purposes without having to install those devices into an existing system first. This makes benchtop testing much easier.

The tester can be powered with a 7-36V input to the onboard Pololu D24V50F5 regulator which will give a constant 5V output at 5A to power the connected devices over the Serial or I2C bus. There is also a +5V output header to power additional devices if needed. Alternately, this header can be used to feed a regulated 5V input to the board if the user does not wish to install the Pololu regulator.

The Command Tester supports sending commands via I2C as well as Serial at the same time. Under normal usage, the command tester does not need to receive I2C commands, however there is receive code in the basic sketch, so it is possible to send I2C commands the tester if the user wishes it. See the Code section below for more information on this.

**Board Versions**

There are currently 3 versions of the Command Tester board. The differences between the boards are:

* V1 board has 15 command buttons, while V1.1 and V1.2 have 14. The reason for this is because Digital pin 13 is connected to the LED on the Arduino Pro Mini and should not be used as an input, therefore it was removed in the later versions.
* The location of the Shift Switch and LED is different on each version of the board.

Other than the above, the boards are functionally equivalent.

A circuit board

Description automatically generatedA circuit board

Description automatically generatedA circuit board

Description automatically generated

V1 V1.1 V1.2

**Arduino Pin Connections**

The Arduino Pins are assigned as follows:

* TX0 Serial Transmit
* RX0 Serial Receive
* D2 Command 1
* D3 Command 2
* D4 Command 3
* D5 Command 4
* D6 Command 5
* D7 Command 6
* D8 Command 7
* D9 Command 8
* D10 Command 9
* D11 Command 10
* D12 Command 11
* D13 On Board LED
* A0/D14 Command 12
* A1/D15 Command 13
* A2/D16 Command 14
* A3/D17 Shift
* A4/D18 SDA (I2C)
* A5/D19 SCL (I2C)

**Code**

The current version of the Command Tester code is located on GitHub (<https://github.com/CBozzoli/Command-Tester/releases>) and consists of two files that must be placed in the same directory:

* Command\_Tester\_Skeleton\_1.1.ino (the main sketch)
* config.h (header file that includes configuration settings)

The sketch is distributed as a “skeleton” to allow users to insert the commands needed for the devices they are testing. This is done by typing the commands into the arrays on the config.h tab. Notes on how to do this are in the sketch.

There are several user configurable settings in the config header:

* Commands
* BAUD Rate (Default is 2400 for Teeces compatibility, 9600 and 57600 are also available for JEDI and Stealth compatibility)
* I2C Address of the Tester (default is 0)
* I2C Addresses for receive devices
* Debug Toggle (turns on/off debug messages)
* I2C Receive Mode (see below for more on this)

The default behavior of the Command Tester is as the I2C bus master and does not need to receive I2C commands; however, the receive code is included in case a scenario arises where it is needed. The receive command function is different depending on what device is sending the I2C commands. For example, the Stealth Controller sends integer I2C commands only, but the Marcduino sends commands as character strings. To deal with this, I have included the SIMPLE\_I2C switch to determine the I2C Receive Mode. If the device sending the commands to the tester is something like the Stealth Controller, then turn on Simple receive by uncommenting this line in the config header. If it is something like the Marcduino, then use the complex receive. A general rule of thumb to help you determine which to use is how you enter the commands in the sketch that is sending them. If you must put the command in quotes (e.g. “10\r”), then use complex. If you do not use quotes (e.g. 10), use simple. The default setting is complex).

Warning: If you do find a need for the tester to receive commands, you may need to do additional coding to accomplish what you need as neither mode will parse JAWALITE commands (both simple and complex modes only handle numerical commands).

**Bill of Materials**

* 1x Arduino Pro Mini
* 1x Pololu D24V50F5 5v, 5A Step-Down Regulator (<https://www.pololu.com/product/2851>)
* Various Male and Female Pin Headers
* 14x Momentary Tactile Button Switch (Digikey Part Number 1528-367-ND, <https://www.digikey.com/products/en?mpart=367&v=1528>)
* 1x 5mm LED
* 1x 1k Ohm Resistor
* 1x Screw Terminal Block (2-Pin, 5 mm Pitch, Side Entry, <https://www.pololu.com/product/2440>)
* 1x 6-pin, 2.54mm Pitch, Self-Latching Switch (I don’t have a part number for these, but the pin layout is below. I purchased them from Amazon.com in a variety pack and used the 8x8mm ones. If you buy these, you should probably test the switch before soldering it on to make sure the pin configuration is correct. The 3rd picture shows proper orientation on the board. <https://www.amazon.com/gp/product/B07J4ZCWNX/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1>).

Diagram, schematic

Description automatically generatedA circuit board

Description automatically generated