

ENEE440F20 Project : 'Testy' Microprocessor I/O Function Tester

What is Testy?

Testy is software running on a NUCLEO-H745ZI-Q microprocessor board. Testy contains a set of 'devices' (initialized components of the 'H745 peripheral set) which are controlled by plain-text messages sent over the board's serial debug port from a PC. Actions specified in the Testy control messages will cause Testy to dispatch different functions on the 'H745 devices which input or output analog or digital values and record or communicate data. Testy will report status and results back to the PC over the serial debug port when requested or on an on-going basis depending on the device function selected.

Moon shot Testy I/O capabilities:

GPIO

- output: immediate, scheduled, repetitive (timed and DMA)
- input: immediate, scheduled, repetitive (timed and DMA)

ADC

- sampling: immediate, repetitive (timed, both interrupt and DMA)

DAC

- output: immediate, repetitive (timed, both interrupt and DMA)

TIM

- output: pulse, pulse-width, pulse-frequency,PWM,PFM,triggered
- input: timed event, event count, pulse frequency, pulse width

RTC

- calender and clock alarms

SPI

- configuration: format, master, slave
- event type: polled, interrupt
- buffering: character, block, ring

I2C

- configuration: format, master, slave
- event type: polled, interrupt
- buffering: character, block, ring

Software for Testy:

Testy requires software running on the PC for testing/demonstrating the 'H745's input and output capabilities:

- hex2Testy ('h2T.exe' in \PC_code\hex2Testy01): program to convert numeric strings into valid Testy 'Cmd' messages.
- testterm ('testterm.exe' in \PC_code\testtermv04): program which sends scripts containing Testy 'Cmd' messages over the PC serial port.
- 03_testy_ws\testy (in 03_testy_ws.zip): Testy green LED 'device' demo.

Testy software on the 'H745 to be written by you and will be based on the provided Eclipse 'testy' project, an updated version of the original Eclipse 'scratchy' project in which the green LED ON/OFF function is controlled by Testy commands sent from the PC.

- your code which interfaces with the hardware is to be written in ARM assembly language. Code for higher level functions (data manipulation, communication protocols, ...) can be library functions or written in C.
- reverse engineering of code generated by magical tools is acceptable, but assembly language drivers for hardware must be reduced to standalone files (no #includes) in well-commented assembly language with no magic numbers for configuration values or register addresses.

(continues on next page)

Documentation:

Your project-report should be submitted in a .zip file and will contain four documents -- readme, user manual, maintainer document, and acceptance test script -- plus the build directory and code.

- readme: lists and *briefly* describes the contents of the .zip submission and directions for use.
- maintainer document (PDF): audience is other engineers who might work on the device either to modify or to service it. It *must* contain a table of contents and an abstract, and should have figures (flow charts, state diagrams, data diagrams, schematics) to assist in the description.
- user's manual (PDF): audience is a technical user who will use the device on a laboratory bench or in the field. It *must* contain a table of contents and operating instructions for all modes of use of the device.
- acceptance test script (.txt): testing will be done using your script of test commands and instructions.
- build directory and code: contains the *well-commented* source and build tools for the device programs and a readme for their use.

Submissions: must be individually identifiable as to source (you) and application (ENEE440F20_PROJECT)