```
#
=======================================================================================
====================================
# Build utility, models RTK_Everywhere GitHub workflow
# last updated: [2024-04-05] @ 05:45pm
# D. Foster (doug@dougfoster.me)
#
# To run:
#   "script rtkb"
# Added to .zshrc
#     alias script=$HOME/Documents/sh-scripts/script.zsh
#   [2024-02-14] for SparkFun RTK project
#   alias rtkdir="cd $HOME/Dropbox/Data/doug/Projects/2024-01-26-
SparkFun_RTK_EVK/"
#   alias scriptdir="cd $HOME/Documents/sh-scripts/"
#
# ZSH conditional expressions - https://zsh.sourceforge.io/Doc/Release/
Conditional-Expressions.html
#
# Repository: https://github.com/sparkfun/SparkFun_RTK_Everywhere_Firmware/
tree/release_candidate
# Modeled after: https://github.com/sparkfun/
SparkFun_RTK_Everywhere_Firmware/tree/release_candidate/.github/workflows/
compile-rtk-everywhere.yml
#
# Following these steps:
#   1) GitHub workflow
#     https://github.com/sparkfun/SparkFun_RTK_Everywhere_Firmware/blob/
main/.github/workflows/compile-rtk-everywhere.yml
#   2) Updating RTK Firmware - Compiling Source - Windows
#     https://docs.sparkfun.com/SparkFun_RTK_Firmware/firmware_update/
#windows_1
#   3) VS Code "tasks.json" from Nathan
#
=======================================================================================
====================================

# -----------
# Set vars.
# -----------

esp32_base_dir=$HOME"/Library/Arduino15/packages/esp32/hardware/
esp32/2.0.11/"  # [2024-03-23] Updated from 2.0.2
sketch_dir=$HOME"/Documents/Arduino/sketches/"
drop_box_dir=$HOME"/Dropbox/Data/doug/"
example_base_dir=$drop_box_dir"Topics/_dev-arduino/SparkFun_u-blox_GNSS_v3
(GitHub)/examples/"
rtk_everywhere_project_dir=$drop_box_dir"Projects/2024-01-26-
SparkFun_RTK_EVK/"
esp32_board="https://raw.githubusercontent.com/espressif/arduino-esp32/gh-
pages/package_esp32_index.json"
esp32_ver="esp32:esp32@2.0.11" # [2024-03-23] Updated from 2.0.2
valid_usage="\nusage: rtkb [module] [branch] [directory] [flags]
  1) module = rtk/test/sk/ex/-
     rtk: ~/Dropbox/Data/doug/Topics/_dev-arduino/
SparkFun_RTK_Everywhere_Firmware (GitHub)
     rtkd: ~/Dropbox/Data/doug/Topics/_dev-arduino/
SparkFun_RTK_Everywhere_Firmware-Doug-Foster (GitHub)
```

```
     test: ~/Dropbox/Data/doug/Topics/_dev-arduino/
SparkFun_RTK_Everywhere_Firmware (GitHub)/Test Sketches
     sk (sketch): ~/Documents/Arduino/sketches/SKETCH_NAME
     ex (example): ~/Dropbox/Data/doug/Topics/_dev-arduino/SparkFun_u-
blox_GNSS_v3 (GitHub)/examples/
     -: not applicable\n
  2) branch = m/rc/-
     m: https://github.com/sparkfun/SparkFun_RTK_Everywhere_Firmware/tree/
main
     rc: https://github.com/sparkfun/SparkFun_RTK_Everywhere_Firmware/tree/
release_candidate
     -: not applicable\n
  3) name = name/-
     name: RTK_Everywhere, directory
     -: not applicable\n
  4) flags = f,d,p,c,u,e
     f (full)
     d (debug)
     p (psram)
     c (compile)
     u (upload)
     e (erase flash)\n
examples:
  script rtkb   -      -   -                                 e
  script rtkb   rtk    m   - (RTK_Everywhere)                f,d,p,c,u
  script rtkb   rtk    rc  - (RTK_Everywhere)
  script rtkb   rtk    -   - (RTK_Everywhere)                u
  script rtkb   rtkd   -   - (RTK_Everywhere-Doug-Foster)    u
  script rtkb   test   -   GNSS_GetPosition_v3               u
  script rtkb   ex     -   directory\n
notes:
  1) Don't forget to toggle repo with 'GitHub Desktop' for 'main' or 'release
candidate' branches'\n"
echo "... done [ Set vars. ] ...\n"

# -----------------
# Validate input.
# -----------------
clear
if [ $# -eq 0 ] ; then
      echo $valid_usage
      exit 1
fi
echo '--- '`date`' ---'
echo 'rtk-build "'$@'" ...'
echo "... check "$#" args ..."
if [ $# -lt 3 ] ; then
      echo "... invalid # of args ..."
      echo $valid_usage
      echo "... quitting."
      exit 1
fi
if ! ( [ $1 = 'rtk' ] || [ $1 = 'rtkd' ] || [ $1 = 'test' ] || [ $1 = 'sk' ]
|| [ $1 = 'ex' ] || [ $1 = '-' ] ) ; then
      echo "... invalid 1st arg (module)..."
      echo $valid_usage
      echo "... quitting."
      exit 1
```

```
fi

if ! ( [ $2 = 'm' ] || [ $2 = 'rc' ] || [ $2 = '-' ] ) ; then
      echo "... invalid 2nd arg (branch) ..."
      echo $valid_usage
      echo "... quitting."
      exit 1
fi
echo "... done [ Validate input. ] ..."

# --------------------------
# Set vars based on input.
# --------------------------

build_module=$1
build_branch=$2
build_name=$3
if [ $1 = 'rtk' ] || [ $1 = 'rtkd' ]  ; then
      build_name='RTK_Everywhere'
fi
build_type='skip'
debug=false
psram=false
compile=false
upload=false
erase=false
if ! [ -z $4 ] ; then  # flags = f,d,p,c,u,e  (full, debug, psram, compile,
upload, erase)
      if [[ $4 == *'f'* ]] ; then
            build_type='full'
      fi
      if [[ $4 == *'d'* ]] ; then
            debug=true
      fi
      if [[ $4 == *'p'* ]] ; then
            psram=true
      fi
      if [[ $4 == *'c'* ]] ; then
            compile=true
      fi
      if [[ $4 == *'u'* ]] ; then
            upload=true
      fi
      if [[ $4 == *'e'* ]] ; then
            erase=true
      fi
fi
echo "... done [ Set vars based on input. ] ..."

# -------------------------------------
# Set repo vars & validate directory.
# -------------------------------------

if [ $build_type = 'full' ] || [ $compile = true ] ; then
      echo "... check for valid repo directory ..."
      # Set repo vars.
      repo_base_dir=$drop_box_dir"Topics/_dev-arduino/
SparkFun_RTK_Everywhere_Firmware (GitHub)/Firmware"
```

```
      repo_dir=$repo_base_dir"/RTK_Everywhere/"
      if [ $1 = 'rtkd' ] ; then
            repo_base_dir=$drop_box_dir"Topics/_dev-arduino/
SparkFun_RTK_Everywhere_Firmware-Doug-Foster (GitHub)/Firmware"
            repo_dir=$repo_base_dir"/RTK_Everywhere/"
      fi
      if [ $1 = 'test' ] ; then
            repo_dir=$repo_base_dir'/Test Sketches/'$3
      fi
      if [ $1 = 'ex' ] ; then
            repo_dir=$example_base_dir$3"/"
      fi
      if [ $1 = 'sk' ] ; then
            repo_dir=$sketch_dir$3"/"
      fi
      # Validate repo directory.
      if ! [ $1 = 'sk' ] ; then
            echo "... "$repo_dir" ..."
            if [ -d $repo_dir ] ; then
                  echo "... repo directory is valid ..."
            else
                  echo "... bad directory name ..."
                  echo "... quitting."
                  exit 1
            fi
      fi
      echo "... done [ Set repo vars & validate directory. ] ..."
else
      echo "... skip [ Set repo vars & validate directory. ] ..."
fi

# -----------------------------------------
# Copy repo dir into Arduino sketch library.
# -----------------------------------------

if [ $compile = true ] ; then
      echo '... module = "'$build_module'", branch = "'$build_branch'" ...'
      echo '... copy "'$build_name'" from repo to sketch directory ...'
      if ! [ $build_module = 'sk' ] ; then
            rm -rf $sketch_dir$build_name
            mkdir $sketch_dir$build_name
            string='cp -R "'$repo_dir'" '$sketch_dir
            if [ $build_module = 'rtk' ] || [ $build_module = 'rtkd' ] ; then
                  string='cp -R "'$repo_dir'" '$sketch_dir$build_name
            fi
            eval $string
      fi
      if [ $build_type = 'full' ] ; then
            echo "... "$build_type" build, run all steps ..."
      else
            echo "... "$build_type" to compile ..."
      fi
      echo "... done [ Copy repo dir into Arduino sketch library. ] ..."
else
      echo "... skip [ Copy repo dir into Arduino sketch library. ] ..."
fi
```

```
#
--------------------------------------------------------------------------------
-------------------------------------
# Reference info
#
# SparkFun Public
#   https://docs.sparkfun.com/SparkFun_RTK_Firmware/firmware_update/
#compiling-source
#   https://github.com/sparkfun/SparkFun_RTK_Firmware/blob/main/.github/
workflows/compile-rtk-firmware.yml
#
# SparkFun Private
#   https://github.com/sparkfun/SparkFun_RTK_Everywhere_Firmware
#   https://github.com/sparkfun/SparkFun_RTK_Everywhere_Firmware/blob/
main/.github/workflows/compile-rtk-everywhere.yml
#
# Arduino
#   https://arduino.github.io/arduino-cli/0.28/configuration/
#   https://arduino.github.io/arduino-cli/0.19/commands/arduino-cli_compile/
#   https://support.arduino.cc/hc/en-us/articles/4415103213714-Find-sketches-
libraries-board-cores-and-other-files-on-your-computer
#
# Locations
#   ~/Library/Arduino15/arduino-cli.yaml
#   ~/Library/Arduino15/inventory.yaml
#   ~/Library/Arduino15/packages/esp32/hardware/esp32/2.0.11 (if core files
are inside a board package)
#   ~/Library/Arduino15/packages/esp32/tools/ (dependencies inside a board
package)
#   ~/Library/Arduino15/staging/libraries (downloadd zip files)
#   ~/Documents/Arduino/hardware/espressif/esp32 (if core files are not
inside a package)
#   ~/Documents/Arduino/libraries
#   ~/Documents/Arduino/skeetches/$sketch/build (compile binaries - set by
$export)
#
# Notes
#   - libraries will not download  if they are up-to-date
#   - compile error: fatal error: SparkFun_IM19_IMU_Arduino_Library.h: No
such file or directory
      #include <SparkFun_IM19_IMU_Arduino_Library.h> //http://librarymanager/
All#SparkFun_IM19_IMU
      # comment out line 36 in RTK_Everywhere.ino
      # // #define COMPILE_IM19_IMU // Comment out to remove IM19_IMU
functionality  #[2024-03-21] D.F.
#   - arduino-cli compile --fqbn "esp32:esp32:esp32":DebugLevel=$
{{ env.DEBUG_LEVEL }} ./Firmware/RTK_Everywhere/RTK_Everywhere.ino
        # --build-property build.partitions=RTKEverywhere
        # --build-property upload.maximum_size=3145728
        # --build-property "compiler.cpp.extra_flags=\"-
DPOINTPERFECT_TOKEN=$POINTPERFECT_TOKEN\" \"-
DFIRMWARE_VERSION_MAJOR=$FIRMWARE_VERSION_MAJOR\" \"-
DFIRMWARE_VERSION_MINOR=$FIRMWARE_VERSION_MINOR\" \"-DENABLE_DEVELOPER=$
{{ env.ENABLE_DEVELOPER }}\""
        # --export-binaries
#   arduino-cli config init --dest-dir . --additional-urls $ESP32BOARD
#   arduino-cli config set library.enable_unsafe_install true --config-file
$ARDUINOCLI_CONFIG
```

```
# ------------------------------------

# --------------------------------
# Create arduino-cli config file.
# --------------------------------

if [ $build_type = 'full' ] ; then
      echo "... create config file for arduino-cli ..."
      arduino-cli config init --additional-urls $esp32_board --overwrite
      arduino-cli config set library.enable_unsafe_install true
      echo "... done [ Create arduino-cli config file. ] ..."
else
      echo "... skip [ Create arduino-cli config file. ] ..."
fi

# ------------------------------
# Update index of cores & lib.
# ------------------------------

if [ $build_type = 'full' ] ; then
      echo "... update index of cores & lib to the latest version ..."
      arduino-cli core update-index
      echo "... done [ Update index of cores ] ..."
      arduino-cli lib update-index
      echo "... done [ Update index of lib. ] ..."
else
      echo "... skip [ Update index of cores & lib. ] ..."
fi

# ------------------------------
# Install ESP32 core version.
# ------------------------------
if [ $build_type = 'full' ] ; then
      echo "... install $esp32_ver board core package & dependent tools ..."
      arduino-cli core install $esp32_ver
      echo "... done [ Install ESP32 core version. ] ..."
else
      echo "... skip [ Install ESP32 core version. ] ..."
fi

# -------------------------
# Change partition table.
# -------------------------

if [ $build_type = 'full' ] && [ $repo_typ = 'rtk' ] ; then
      echo "... change partition table ..."
      source=$repo_base_dir'/RTKEverywhere.csv' # was "app3M_fat9M_16MB.csv"
      destination=$esp32_base_dir'tools/partitions/'
      echo "... copy "$source" ..."
      echo "... to "$destination" ..."
      cp $source $destination
      echo "... done [ Change partition table. ] ..."
else
      echo "... skip [ Change partition table. ] ..."
fi

# ------------------------------------
# Install 'Arduino approved' libraries.
```

```
# ----------------------------------------

if [ $build_type = 'full' ] ; then
      echo "... install known libraries ..."
      arduino-cli lib install ArduinoJson@6.19.4
      arduino-cli lib install ESP32Time@2.0.0
      arduino-cli lib install ESP32_BleSerial@1.0.4
      arduino-cli lib install "ESP32-OTA-Pull"@1.0.0 --no-deps  # skip newer
version of ArduinoJson
      arduino-cli lib install Ethernet@2.0.2
      arduino-cli lib install JC_Button@2.1.2
      arduino-cli lib install PubSubClient@2.8.0
      arduino-cli lib install "SdFat"@2.1.1
      arduino-cli lib install "SparkFun LIS2DH12 Arduino Library"@1.0.3
      arduino-cli lib install "SparkFun MAX1704x Fuel Gauge Arduino
Library"@1.0.4
      arduino-cli lib install "SparkFun u-blox GNSS v3"@3.0.14
      arduino-cli lib install SparkFun_WebServer_ESP32_W5500@1.5.5
      # arduino-cli lib install "SparkFun Qwiic OLED Arduino Library"@1.0.9
#Already added by (which?) dependency?
      arduino-cli lib install "SparkFun Qwiic OLED Arduino Library"@1.0.13  #
[2024-03-21] Added.
      arduino-cli lib install SSLClientESP32@2.0.0  # [2024-02-17] Added.
      # arduino-cli lib install "SparkFun UM980 Triband RTK GNSS Arduino
Library"  # [2024-02-17] Added. [2024-03-21] Removed.
      arduino-cli lib install "SparkFun Extensible Message Parser"@1.0.0  #
[2024-02-17] Added.
      arduino-cli lib install "SparkFun BQ40Z50 Battery Manager Arduino
Library"@1.0.0  # [2024-02-17] Added.
      arduino-cli lib install "ArduinoMqttClient"@0.1.8  # [2024-02-17]
Added.
      arduino-cli lib install "SparkFun 9DoF IMU Breakout - ICM 20948 -
Arduino Library"  # [2024-02-17] Added.
      echo "... done [ Install 'Arduino approved' libraries. ] ..."
else
      echo "... skip [ Install 'Arduino approved' libraries. ] ..."
fi

# ----------------------------------------
# Install 'Arduino NOT approved' libraries.
# ----------------------------------------

if [ $build_type = 'full' ] ; then
      echo "... install unapproved libraries for sketches ..."
      # arduino-cli lib install --git-url https://github.com/sparkfun/
SparkFun_Qwiic_OLED_Arduino_Library.git  # [2024-03-21] Removed.
      arduino-cli lib install --git-url https://github.com/sparkfun/
SparkFun_u-blox_PointPerfect_Library.git  # [2024-03-21] Added.
      arduino-cli lib install --git-url https://github.com/sparkfun/
SparkFun_Unicore_GNSS_Arduino_Library.git # [2024-03-21] Added.
      arduino-cli lib install --git-url https://github.com/me-no-dev/
ESPAsyncWebServer.git
      arduino-cli lib install --git-url https://github.com/me-no-dev/
AsyncTCP.git
      if [ build_module = 'test' ] ; then
            arduino-cli lib install --git-url https://github.com/LeeLeahy2/
SdCardServer.git  # WebServer
      fi
```

```
      echo "... done [ Install 'Arduino NOT approved' libraries. ] ..."
else
      echo "... skip [ Install 'Arduino NOT approved' libraries. ] ..."
fi

#
# ------------------------------------------------------------------------
# Install Python serial library (needed by esptool otherwise compile fails).
#
# ------------------------------------------------------------------------

if [ $build_type = 'full' ] && [ $build_module = 'rtk' ] ; then
      echo "... install python pyserial library ..."
      pip3 install pyserial
      echo "... done[7] ...\n"
      echo "... done [ Install Python serial library. ] ..."
else
      echo "... skip [ Install Python serial library. ] ..."
fi

# ----------------------------------------
# Python script - update form.h web file.
# ----------------------------------------

if [ $build_type = 'full' ] && [ $build_module = 'rtk' ] ; then
      echo "... update RTK_Everywhere/form.h using python ..."
      START_DIR=$PWD
      echo "... change to RTK_Everywhere Tools directory ..."
      cd $repo_base_dir'/Tools'
      echo "... run python script index_html_zipper.py - update
index_html ..."
      python3 index_html_zipper.py ../$build_name/AP-Config/index.html ../
$build_name/form.h
      echo "... running python script main_js_zipper.py - update main_js ..."
      python3 main_js_zipper.py ../$build_name/AP-Config/src/main.js ../
$build_name/form.h
      echo "... return back to script directory ..."
      cd $START_DIR
      echo "... done [ Python script - update form.h web file. ] ..."
else
      echo "... skip [ Python script - update form.h web file. ] ..."
fi

# ----------
# Compile.
# ----------

sketch=$sketch_dir$build_name"/"$build_name".ino"
upload_dir=$rtk_everywhere_project_dir'_builds/_upload/'
if [ $compile = true ] ; then
      echo "... setting compile options ..."
      echo "... sketch is "$sketch" ..."
      echo '--- '`date`' ---'
      DATE=$(date +'20%y-%m-%d@%H-%M-%S')

destination_dir=$rtk_everywhere_project_dir'_builds/'$DATE'**'$build_name'/'
      source_dir=$sketch_dir$build_name'/build/esp32.esp32.esp32/'
      #fqbn='esp32:esp32:esp32'
```

```
      #fqbn="--fqbn esp32:esp32:esp32:PSRAM=enabled,DebugLevel=debug"
      #--optimize-for-debug
      # eval arduino-cli board attach -b esp32:esp32:esp32 -p /dev/
cu.usbserial-14130 $sketch
      # fqbn="--fqbn esp32:esp32:esp32:UploadSpeed=115200:PSRAM=enabled"
      if [ $psram = true ] ; then
            fqbn="--fqbn esp32:esp32:esp32:PSRAM=enabled"
      else
            fqbn="--fqbn esp32:esp32:esp32"
      fi
      if [ $debug = true ] ; then
            fqbn=$fqbn",DebugLevel=debug"
            echo '... include compile option "debug" ...'
      fi
      echo "... beginning compile, be patient ..."
      build_prop_1="--build-property build.partitions=app3M_fat9M_16MB"
      build_prop_2="--build-property upload.maximum_size=3145728"
      build_prop_3='--build-property "compiler.cpp.extra_flags=\"-
DENABLE_DEVELOPER=true\""'
      export="--export-binaries"
      command="$fqbn $sketch $build_prop_1 $build_prop_2 $build_prop_3
$export $push"
      eval arduino-cli compile $command
      echo "... end of compile ..."
      echo '--- '`date`' ---'
      if [ -d $sketch_dir$build_name"/build" ] ; then
            echo "... copy binaries to build directories ..."
            mkdir $destination_dir
            echo "... saving compile notes ..."
            notes=$destination_dir"compile-config.txt"
            touch $notes
            echo "arduino-cli compile "$command >> $notes
            echo "\n" >> $notes
            cp $source_dir* $destination_dir.
            echo "... copied binaries (all) from sketch dir to build archive
dir ..."
            # [2024-03-25] Bins from compile don't work, why?
              # cp "$source_dir$build_name"'.ino.bootloader.bin'
"$upload_dir$build_name"'.ino.bootloader.bin'
              # cp "$source_dir$build_name"'.ino.partitions.bin'
"$upload_dir$build_name"'.ino.partitions.bin'
            cp "$source_dir$build_name"'.ino.bin'
"$upload_dir$build_name"'.ino.bin'
            echo "... copied binaries (firmware) from sketch dir to upload
dir ..."
            final_dir=$destination_dir # Rename archive directory to be more
descriptive.
            if [ $build_module = 'rtk' ] || [ $build_module = 'rtkd' ] ; then
                  final_dir=${destination_dir::-1}'_'$build_branch
                  if [ $build_module = 'rtkd' ] ; then
                        final_dir=$final_dir'-Doug-Foster'
                  fi
                  mv $destination_dir $final_dir
            fi
            echo '... bin directory is "'$final_dir'" ...'
      fi
      echo "... done [ Compile. ] ..."
else
```

```
      echo "... skip [ Compile. ] ..."
fi

#
--------------------------------------------------------------------------------
-----------------------------------
# Reference info
#
# https://docs.espressif.com/projects/esptool/en/latest/esp32/
# https://github.com/sparkfun/SparkFun_RTK_Firmware_Uploader
# https://docs.sparkfun.com/SparkFun_RTK_Firmware/firmware_update/
#
# If building with Arduino IDE, set Tools:
      # A. 'Flash Size' to '16MB (128Mb)''
      # B. 'Partition Scheme' to '16M Flash (3MB APP/9MB FATFS)'
            # This will cause the build use the 'app3M_fat9M_16MB.csv'
updated partition table.
            # Needed since the full EVK source is so large.
      # C. 'PSRAM' to 'Enabled'
      # D. 'Upload Speed' to '115200'
#
# Upload new firmware:
      # Option 1: Over-The-Air (WiFi - serial menu)
      # Option 2: RTK uploader (serial)
      # Option 3: SD card
            #updating-firmware-using-the-uploader-gui
      # Option 4: WiFi
            #updating-firmware-from-wifi
      # Option 5: Command line interface (CLI)
            # esptool.exe (Windows), python esptool.py (MacOS/Linux)
            # esptool.py
            #updating-firmware-from-cli
                  # esptool.exe --chip esp32 --port COM6 --baud 921600 --
before default_reset --after hard_reset write_flash -z --flash_mode dio --
flash_freq 80m --flash_size detect 0x1000 ./bin/
RTK_Surveyor.ino.bootloader.bin 0x8000 ./bin/
**RTK_Surveyor_Partitions_4MB**.bin 0xe000 ./bin/boot_app0.bin 0x10000 ./
RTK_Surveyor_Firmware_vxx.bin
            # copied bootloader, partition, & boot_app from
                  # '/Applications/RTK Uploader/RTKUploader.app/Contents/
Frameworks/resource/
#
--------------------------------------------------------------------------------
-----------------------------------

command=''
chip='--chip esp32'
port='--p /dev/cu.usbserial-1424120'
speed='--baud 460800'  # 115200, 460800, 565020, 921600

# ---------
# Upload.
# ---------
if [ $upload = true ] ; then
      echo '... upload to rtk ...'
      before='--before default_reset'
      after='--after hard_reset'
      # operation='write_flash -z'
```

```
      operation='write_flash'
      flash_mode='--flash_mode dio'
      flash_freq='--flash_freq 80m'
      flash_size='--flash_size detect'
      boot_app='# 0xe000 '$upload_dir'boot_app0.bin'  # Not used?
      # Needed if flash is erased. Always send to be safe.
        boot_loader='0x1000 '$upload_dir'RTK_Surveyor.ino.bootloader.bin'
        partitions='0x8000 '$upload_dir'RTK_Surveyor_Partitions_16MB.bin'
      # [2024-03-25] Bins from compile don't work, why?
        # boot_loader='0x1000 '$upload_dir'$build_name.ino.bootloader.bin'
        # partitions='0x8000 '$upload_dir'$build_name.ino.partitions.bin'
      firmware='0x10000 '$upload_dir$build_name'.ino.bin'

      command="esptool.py $chip $port $speed $before $after $operation
$flash_mode $flash_freq $flash_size $boot_loader $partitions $firmware"
      echo '--- '`date`' ---'
      echo '... beginning upload ...'
      echo '... '$command' ...\n'
      eval $command
      echo '... upload complete ...'
      echo '--- '`date`' ---'
      echo "... done [ Upload. ] ..."
else
      echo "... skip [ Upload. ] ..."
fi

# --------------
# Erase flash.
# -------------

if [ $erase = true ] ; then
      echo '... erase flash ...'
      command="esptool.py $chip $port $speed erase_flash"
      eval $command
      echo "... done [ Erase flash. ] ..."
else
      echo "... skip [ Erase flash. ] ..."
fi

# ----------
# Finished
# ----------

echo '... finished!'
echo '--- '`date`' ---\n'
```