Long Mission Mode for lower energy use

1/25/20 Brent Roman brent@mbari.org

Traditional ESP Missions

- A sequence of “phases”, each with a prescribed start time
  - Actions predetermined by puck load

- ESP sleeps between phases. While “asleep”:
  - Still monitors contextual sensors
  - Still maintains radio context with shore

- All phases began at times prescribed in the mission script
  - Start times specified may be absolute or relative
    - Relative times specify the “sleep time” between phases

- Adaptive sampling may start any phase early

“Long” mode ESP Missions

- ESP's Linux Host is powered off between phases
  - Only the “sleepy” microprocessor remains powered
  - Consumption drops to ~100mW! (+ ~150mW for modem)

- ESP drops off the network when its Linux Host is powered off
  - Deployment data may still be available from shore server
  - Some radio modems allow a “wake up” call to power up host
    - Our Sierra Wireless modems draw ~150mW in “standby”

- All mission phases start exactly at their prescribed times
  - Adaptive Sampling via Triggers is **NOT** possible
  - CTD runs in its logging mode (no ISUS support)

Mission Script format is unchanged :-)

- Mission Scripts in long mode are reread on each wakeup
  - Every time the ESP wakes, it does a full system reboot
  - Reprocesses the entire mission each time!
  - A small text file records mission progress
    - Allows ESP to skip the appropriate number of phases
- Restrictions:
  - Any actions coded between phases run on every restart.
  - No monitoring of context, etc. between phases.
  - Computations on phase start times may be troublesome, as they will be rerun on every restart!
  - No referencing variables computed in previous phases.

Mission Interactions

- Sends status email on power up and power down
- espclient via ssh works as usual
  - FTP and HTTP work as well
- Each status email triggers data upload to shore server
  - ESP may be off-line at the top of each hour
  - On shutdown, the latest data may not yet be uploaded.
    - A wakeup call can be used to force another upload.

Error Handling

- Any unhandled exception still generates an email
  - And suspends the failed thread
- Remains powered only if ESP mechanism was “in use”
  - Because pucks may be in play, etc.
  - Otherwise, mission enters a “paused” state
- ESP powers down after 8 minutes in paused state, unless:
  - User resumes suspended thread,
  - Issues a command to stay paused indefinately, or
  - Exits the ESP application

Wake Up Calls

- Our cell modem is a phone with a normal phone number
- Calling it while it is not active causes it to send the text
  - RING RING RING RING …
- The Sleepy microcontroller watches for this text.

- To wake up a “dead” ESP, call the modem's number
  - let it ring 4 (or more) times, then hang up.

- 90 seconds later, ESP should be on-line.
  - But, remember, it will be “paused”, so...
  - It will power down in 8 minutes unless you prevent it.

Reset and Power Cycles

- Reset and Power Failures will **not** terminate long missions
  - Provided they occur between active phases
- They are handled like Wake Up Calls
  - Mission pauses for 8 minutes and continues.
- RS-232 BREAK from the modem causes sleepy to simulate a hardware reset.
  - But this only works while Linux host is powered down.
- Reboots during mission phases cause mission to suspend
  - Generates an error email
  - Mission may be resumed after operator intervenes
    - Recovery may be challenging...

Modifying Long Missions as they run

- Long Mission scripts are reread each time the ESP wakes
- So, while a phase is being processed
  - You may *carefully* edit the “running” mission script file
  - Change start times and actions in future phases only
  - Take care not to alter history
    - Any change in phases that already ran...
      - Will cause a rip in the fabric of time-space! :-)
      - and an error on the next reboot
- Beware that ESP will powerdown when current phase ends
  - The stayUp command will prevent this

Simulating Long Missions

- Long Missions simulate powering down the ESP by...
  - exiting the ESP simulator after each phase completes

- You must repeatly simulate until entire mission completes
  - Or use...
- New simfaster and quicker ESPmodes for long missions:
  - Correspond to respective simfast and quick modes
  - But, they simulate entire mission, not each phase.
    - These modes work by setting the “$stayUp=true”

Enabling Long Mission Mode

- In phasecfg.rb, require 'longmission' instead of 'mission'
  - There is no sensorPolling in long missions
    - Items in phasecfg that normally configure it are ignored
  - Example phasecfg.rb for long missions found at:
    - $ESPhome/mission/long/phasecfg.rb
- Putting esp2/mission/long first on the $ESPpath
  - Will use 'long' phasecfg.rb without altering standard one
- To make espserver start at system boot:
  - Add espserver to directory of boot-time startup scripts
    - By adding a simlink to it in /etc/rc.d/rc3.d
- To make ESPserver continue your mission at boot:
  - Alter mission= line in $ESPhome/bin/espserver

Phase File

- The file /var/log/real.phase records the progress of the mission as phases execute. Actual name is /var/log/${ESPmode}.phase

  ESP waldo

  MISSION /home/brent/esp2/mission/long/archives.rb

  STARTED 16:00:00.06PST31-Dec-69

  PHASE 5:45PM 4/20/16 {WCR 100} #13

  AWAIT 17:45:00.00PDT20-Apr-16

  BEGUN 17:45:26.72PDT20-Apr-16

  ENDED 18:10:21.56PDT20-Apr-16

  PHASE 9PM {WCR 101} #17

  AWAIT 21:00:00.00PDT20-Apr-16

  BEGUN 21:00:26.72PDT20-Apr-16

  ENDED 21:25:35.86PDT20-Apr-16

  ACCOMPLISHED 06:25:43.33PDT21-Apr-16

Long Misson Commands

- stayPaused
  - cancels powerdown when ESP is in “paused” state
- stayUp
  - ESP stays UP when it would normally powerdown.
  - stayUp false will cancel this.
- missionLog
  - Displays all the phases of the running mission
- reset! #similar to forgetESPstate OS command on next slide
  - Forget the current mission (start over!)
- ESP.powerdown time
  - Powerdown immediately and power up at specified time
    - Remain off if time omitted (but, still respond to wakeup calls!)
    - time may also be a number of seconds to remain powered down
- ESP.powerdown! time
  - Like ESP.powerdown, but works even if $stayUp is set

How to Forget State

- Long mission mode will remember state after mission ends
  - So reboots don't restart the finished mission!
- You must clear out the .phase file to start a new mission
- Linux command:
  - $ forgetESPstate #removes .phase and .puck files
- Affects simulated missions, too!
  - ESPmode=quick forgetESPstate #for quick sim state
  - ESPmode=simfast forgetESPstate #for simfast sim
- Use the forgetESPstate if
  - ESP complains that mission has already ended
  - Or that new mission does not match the current one.