Dreaming on Mars: How Curiosity Performs Actuator Warm-Up while Sleeping

Gene Y. Lee
Systems Engineering Section
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA, U.S.A.
Gene.Y.Lee@jpl.nasa.gov

James A. Donaldson
Flight Electronics & Software Systems Section
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA, U.S.A.
James.A.Donaldson@jpl.nasa.gov

Abstract *- Before the Curiosity rover can perform its science activities for the day, such as driving, moving its robotic arm, or drilling, it first has to ensure that its actuators are within their allowable flight temperatures (AFTs). When the rover is awake, flight software uses heaters to warm up and maintain thermal zones at operational temperatures. However, Curiosity spends about 70% of its time sleeping, with the flight computer off, in order to conserve energy. Dream Mode is a special behavior that allows the rover to execute warm-up activities while sleeping. Using Dream Mode, actuators can be warmed up to their AFTs before the flight computer wakes up and uses them – saving power and improving operational efficiency. This paper describes the motivation behind Dream Mode, how it was implemented and tested on Curiosity, and the challenges and lessons learned along the way.*

Keywords: Dream Mode, Curiosity, Mars, sleep, thermal, fault protection.

1 Rover Overview

The Curiosity rover landed in Gale Crater on Mars in August 2012. With an impressive suite of science instruments, the rover's mission to investigate the habitability of Mars is now underway. A description of the rover avionics (and acronyms) and how they are used with respect to wakeup and shutdown behavior will provide a basis for understanding Dream Mode.

The supporting infrastructure behind Curiosity's machinery lies in dual PAM (Power and Analog Module) assemblies. Each PAM consists of a REU (Remote Engineering Unit) and four LCCs (Load Control Cards). The REU FPGA (field-programmable gate array) uses a state machine to manage the wake-sleep cycle that Curiosity follows and has the ability to measure engineering data throughout the rover. Each REU also contains blocks of programmable logic called personality PROM (PPROM) that is stored in non-volatile EEPROM memory. The PPROM code is responsible for turning on and off loads upon transitions in the REU state machine. The LCCs allow for power switch control of loads on the rover such as actuators, heaters, instruments, and the RCE (Rover Compute Element) assembly. The RCE, the operational brain of Curiosity, executes flight software (FSW) on a RAD750 computer to orchestrate science and engineering activities while the rover is awake.

When the rover shuts down, the FSW sets alarm clocks in the REU and commands a transition from the awake state to the sleep state. This transition triggers a block of PPROM code that turns off the RCE. While sleeping, the alarm clocks count down until they expire. The expiration event causes the REU to transition out of the sleep state – triggering another block of PPROM code that turns the RCE back on. When the FSW boots from the RCE, the rover is now back in the awake state and ready to perform the activities planned for the day.

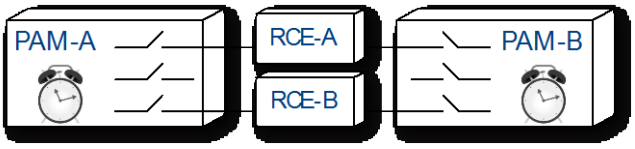

Figure 1. Wakeup and Shutdown Hardware

Figure 1 shows a simplified block diagram of the PAMs, the LCCs (generalized by three switches), and the RCEs. The clock represents the REU state machine and the PPROM code that turns the RCE switch on and off upon waking up and shutting down.

2 Dream Mode Purpose and Evolution

Curiosity's ability to effectively perform science on the surface of Mars is directly related to the amount of energy that is available. The rover is powered by a MMRTG (Multi-Mission Radioisotope Thermoelectric Generator) that produces roughly 114W of power. The power from the MMRTG is stored in two rechargeable lithium-ion batteries. At around 50W, the RCE draws a considerable amount of power while the rover is awake – almost half of what the RTG produces. This, combined with other operational loads, results in the rover having to draw

power from the batteries when the FSW is running. While sleeping, with the flight computer off, the energy demands for Curiosity decrease to the point where the batteries can be recharged. The wakeup and shutdown cycles necessary to manage the energy available for science was understood at the beginning of the rover design. However, developments from the actuator area caused the engineering team to re-evaluate the energy and wake-sleep situation.

Initial work on the energy front was based on an assumption that actuators could be used immediately after the RCE was turned on. The actuator designs were originally based on using "dry-lube" materials that could provide lubrication at the low temperatures experienced on the Martian surface. Ultimately, the "dry-lube" designs did not pass the required tests and alternate designs were proposed using more traditional lubricants. These lubricants had viscosities that changed with temperature, requiring actuators to be pre-heated prior to use. The only way to turn on actuator heaters to perform the pre-heat was to wake up the rover and command the FSW to turn them on.

If the RCE remained powered on throughout the warm-up activity until the actuators achieved the desired temperature, then the amount of available energy for science would be significantly reduced. On the other hand, if the rover put itself to sleep while the heaters warmed up the actuators in an open-loop fashion, then the possibility existed that the heaters could get too hot and damage the actuators (or the heaters themselves). Scheduling multiple wakeup and shutdown cycles during the pre-heat to avoid damage to the actuators would power cycle the RCE repeatedly, using up additional power and adding a threat of thermal stress.

The energy costs and rover safety concerns associated with actuator heating drove the project to add functionality that would address these issues. The engineering team began to pursue a solution that allowed for the pre-heat of actuators and maintenance of actuator temperatures within safety limits while the rover was sleeping. The new functionality would also be capable of responding to faults while sleeping and collecting diagnostic telemetry overnight. Given the impending launch date, the only place to address these issues was by modifying the behavior of the delivered PAMs.

2.1 Hardware Modifications

The original REU FPGA design used a custom micro-code instruction set comprised of six opcodes. These instructions were focused on supporting the primary functions of the REU: allowing the wake-sleep state machine to power on and power off specific loads and allowing explicit control over how analog data from the rover was sampled. As such, the REU was capable of controlling switches upon state transitions and gathering telemetry while the rover was in the sleep state, but the REU did not have the ability to intelligently operate on the data that was collected.

Given schedule constraints and the large amounts of testing already invested in the current design, pursuing a more generalized micro-processor based implementation was not practical. When evaluating the requirements for warming up actuators, gathering data, and performing fault protection while sleeping, the hardware designers realized that these functions could be achieved by modifying the existing design within the architecture already supported by the FPGA.

The lead designer proposed the creation of a new state in the FPGA called "Dream Mode" that would enable the REU to invoke a new engine that would execute specially designed programs while in the sleep state. Five additional opcodes were added to the instruction set to perform arithmetic operations and to load a custom sleep-mode measurement table.

With these additions, the engineering team was now capable of developing code to pre-heat selected actuators while sleeping (the actuators would be at their operational temperatures when the rover woke up), collect diagnostic telemetry from the REU, and monitor the health of the rover – all with the RCE off.

3 REU and FSW Implementation

The FPGA modifications in the REU provided the hardware tools that allowed the Dream Mode functions to be performed. The implementation of these behaviors into operational software was manifested in PPROM code in the REU, FSW in the RCE, and the interfaces between them.

3.1 Dream Mode Code

A PPROM code deliverable was developed that combined the new opcodes into functional Dream Mode behaviors. The set of commands and instructions that constitute the Dream Mode code are stored in REU EEPROM. There are two blocks of code currently on the rover: a fault protection program and a thermal control program.

3.2 Dream Mode Fault Protection

Every time Curiosity shuts down, Dream Mode collects telemetry from PRTs (platinum resistance thermometers) and the REU, stores the values in volatile SRAM memory, performs fault checks on the data. These actions are continuously executed at 5 minute intervals while the rover is sleeping.

REU-specific fault checks include ensuring that the alarm clocks are unmasked (only way of waking up the rover), verifying the REU state machine is functioning as expected, and checking that voltage levels and thermostat switch states are as expected.

With the ability to turn on actuator heaters while sleeping, Dream Mode is responsible for protecting against faults such as switches that can stick on. With a stuck-on switch, a heater can overheat and potentially damage an actuator. To address this risk, the fault protection code evaluates PRT telemetry against too-hot fault limits. If the limit is exceeded, error handling code captures diagnostic data and calls a fault response. Dream Mode runs a single fault response: wake up the rover, turn on the RCE, and leave breadcrumbs to allow the FSW to diagnose and correct the problem.

3.3 Thermal Warm-Up Control

Due to the large size of the actuators that Curiosity uses to move its many mechanisms, PRTs and heaters could not be placed directly on the area they are attempting to control. Thermal models were developed to predict the temperature PRTs would have to reach in order to be sufficiently confident of the internal temperature of the actuators. These models were used to generate warm-up profiles where heaters were turned on for sufficiently long durations to thermally soak the actuator and then duty cycled such that the temperature would be maintained.

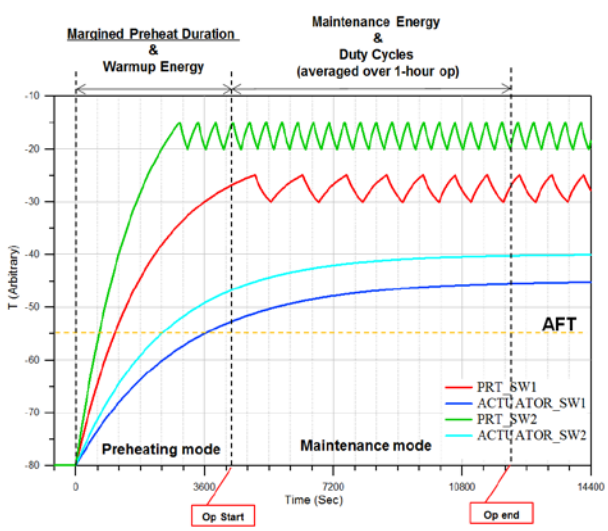

Figure 2. Actuator Warm-Up Profile

Figure 2 shows the warm-up profiles for two actuators, where the pre-heat and maintenance activities have to control the PRTs at significantly higher temperatures than the actuators themselves – in order to ensure they are above the AFT limit.

While the rover is awake, actuator warm-up activities are controlled by FSW commands. These commands specify the thermal zone containing the actuator, the activity start time and duration, and the pre-heat and maintenance temperature limits. The pre-heat duration characterizes the initial ramp-up to AFT where the actuator heater is continuously on. The operational activity is allowed to begin after a successful pre-heat, where the heater is placed in maintenance mode and power cycled as necessary for the actuator to remain above AFT.

This rather complex, actively-controlled thermal FSW design had to be translated into behavior that Dream Mode could perform while sleeping. The FSW schedules all of the warm-up activities (both awake and sleep) for the rover in a thermal table. Upon shutting down, the FSW uses the information in the table to pass up to 30 entries for Dream Mode to execute in the form of Master Control Tables (MCTs). These warm-up activities point to the thermal control PPROM code (Dream Mode Execution Table, or DMET) and use timers that initiate a pre-heat at the activity start time and then evaluate, every 5 minutes, whether to turn on or off a heater based on a control PRT and temperature limits.

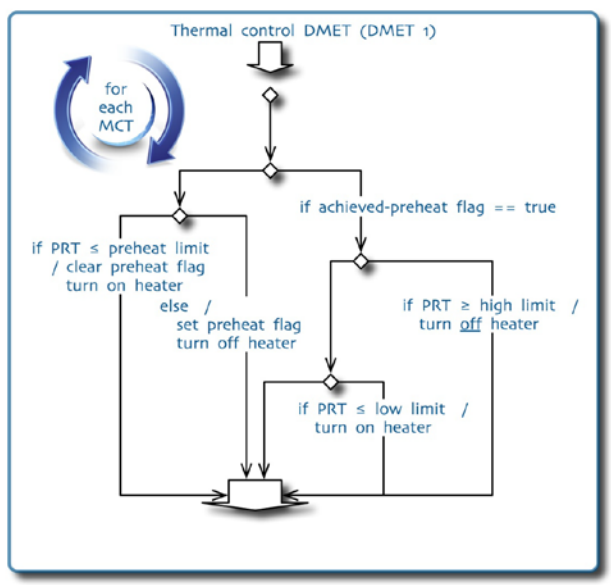

Figure 3. Dream Mode Thermal Control Algorithm

The warm-up algorithm implemented by Dream Mode is shown in Figure 3. While Dream Mode is active, the PPROM code populates SRAM with diagnostic data, PRT temperatures, and pre-heat status. A pre-heat achieved status flag is created for each table entry that corresponds to a thermal zone.

While the rover is sleeping, Dream Mode is executing both fault protection and thermal control at 5 minute intervals, controlling switches, gathering telemetry, and

updating the pre-heat status as necessary. When the rover wakes up, the FSW reads the data out of SRAM and reports on activity failure or success, and then transitions to traditional FSW control.

3.4 Overnight Temperature Sampling

Prior to Dream Mode, the only way to collect temperature data during the Martian night would be to periodically wake up the rover, gather telemetry, and then shut down again. This overnight data would be used to fill in temperature profile gaps and to check and re-correlate thermal models. With the sleep functionality provided by Dream Mode, these night-time wake ups no longer need to be scheduled, resulting in simplified operations and energy savings from not having to turn on the RCE.

While sleeping, Dream Mode collects data from 78 PRTs and stores the time-tagged values in a SRAM ring buffer every hour. The buffer is sized to allow for up to 24 hours of temperature collection until the values are overwritten. Upon waking up, the FSW collects the data from the ring buffer, creates a data product with the values, and populates thermal data channels with the overnight values.

4 Test History

The verification and validation of the Dream Mode functionality occurred in several test phases, beginning with the testable assertions of the hardware modifications in the REU FPGA. The testable assertions defined the basic, functional behaviors the REU would have to demonstrate in order to prove the FPGA design was capable of implementing the desired capabilities. Tests were developed to verify the additional opcodes, registers, and timers behaved as expected. Notional thermal control scenarios tested the ability of the REU to collect PRT telemetry while sleeping and use the values to turn switches on and off. Fault protection cases were divided into two error types: intrinsic and extrinsic errors. Intrinsic error cases tested the FPGA's response to configuration faults in the REU, such as memory corruption in the EEPROM. Extrinsic errors characterized problems external to Dream Mode, such as switch failures and overheat conditions. Regression tests for previous testable assertions were executed as well to show that the new features were non-interfering with the existing REU functionality.

When the requirements were finalized and the algorithms were implemented, dedicated PPROM tests were developed to test all of the error cases, fault checks, and PRT/switch combinations. Specialized scripts were created to write warm-up activities to the REU SRAM, modeling the passage of data from the FSW to the REU upon the transition to sleep without the RCE ever being on. By separating the RCE and FSW behaviors from these tests, REU specific test cases were capable of being scripted to rapidly execute in succession.

The FSW half of the Dream Mode functions were coded in primarily the wakeup/shutdown and thermal software modules. Wakeup and shutdown requirements defined how the FSW passed and read data to and from the MREU upon shutting down and booting up. These tests primarily consisted of parameter manipulation, such as creating warm-up table entries for every possible PRT and heater combination for nominal cases, and setting every PRT and heater sick for off-nominal cases.

The thermal portion of the FSW implementation was more challenging to test. Previously, all warm-up commands were expected to execute with the RCE on and FSW in control. New commands were created to give operators the ability to schedule warm-up activities overnight using Dream Mode. These commands captured warm-up activities in a non-volatile table that persisted across boot cycles, allowing data to be passed from the table to the REU upon the transition to sleep. As with any new development, initial FSW deliveries of the thermal module contained numerous bugs and further testing had to be postponed until they were fixed.

The functional capabilities of the REU PPROM and FSW modules were tested in isolation from one another in order to build sufficient confidence in the individual parts. While this methodology provided for more efficient verification, it was the integration of all these behaviors in a scenario test that would demonstrate how valid the Dream Mode capability actually was.

End-to-end scenario tests were constructed to rigorously test the REU and FSW interfaces upon the wakeup and shutdown boundary. These tests consisted of scheduling warm-up activities during sleep with the rover awake, FSW passing thermal data to the REUs upon shutting down, the execution of Dream Mode thermal control with the rover sleeping, and FSW reading the pre-heat status upon waking back up. While the typical implementation of Dream Mode is to begin warming up a zone while sleeping and waking up just after the pre-heat completes, there are many other combinations of the awake-sleep-awake behavior that were tested. These included warm-up activities that began before shutting down, persisting throughout the entire sleep duration, and continuing while waking back up. Additionally, tests included failure permutations where the pre-heat could fail either during sleep (Dream Mode) or while awake (FSW).

At the PPROM level, intrinsic and extrinsic error cases were only tested to verify that the hardware response to a fault was to turn on the RCE. Scenario testing, by combining both the REU and FSW aspects of Dream Mode, went the next step to verify the appropriate software

response was called when the FSW booted from the RCE. These end-to-end fault protection tests uncovered a few issues that would have not been as obvious had the tests been executed separately, such as recognizing that the warm-up table should be cleared in response to a Dream Mode fault.

5 Operational Usage

The Dream Mode functionality has been disabled on the rover since it landed in August 2012. All of the files necessary to support Dream Mode activities were loaded onto the rover when updating the FSW to the surface version – about a week into surface operations. A couple of hundred sols later, Dream Mode remains disabled for two primary reasons.

One, the thermal environment of the rover is warmer than what was predicted, due to very conservative thermal models and heat radiation from the RTG. Curiosity landed in the spring season of Mars and is currently in summer – as a result, many actuators require very little or no warm-up heating at all. These favorable conditions have resulted in no driving need to start warm-up heating activities while sleeping. The other reason is the current software version is not robust enough to execute Dream Mode behavior. Because the development of the FSW related to Dream Mode was pushed back until after launch and landing, testing, discovering bugs, and fixing them slipped to the right as well. Flight rules are in place to describe the limitations when using Dream Mode in the current software version. A software update planned for 2013 will have the latest fixes and should be capable of running Dream Mode without any restrictions.

A first time activity to use Dream Mode is planned to be executed before this software update. This activity will not perform actuator warm-up while sleeping in support of operations, but will instead focus on demonstrating the basic functions of Dream Mode, such as telemetry collection and notional heating scenarios. These functions will educate the surface operations team on Dream Mode usage, implementation, and the telemetry required to monitor its execution.

During the summer and autumn seasons, a mix of warm-up activities are planned to occur while the rover is awake and while it is sleeping using Dream Mode, depending on the nature of the activity and the operations timeline. When Curiosity enters the winter season at the end of the year where temperatures are expected to drop significantly, Dream Mode will be relied on more heavily to schedule additional and longer duration actuator warm-up activities while sleeping – saving energy and maximizing the time available for science.

6 Challenges and Lessons Learned

As the Dream Mode functionality was developed, implemented, and tested, several challenges were discovered and overcome.

6.1 Testbed Issues

Multiple testbeds were assembled in order to test the hardware and software functionality of Curiosity's systems. As a result of hardware availability and fidelity considerations, each testbed was not always complete and capable of performing the test given to it. Only one testbed, the Mission Systems Testbed (MSTB), would usually have dual PAMs and RCEs. Additionally, it was the only testbed that had a temperature simulator connected to the PRTs in the PAMs (ability to set temperatures). Given these constraints, Dream Mode tests were confined to run on the MSTB only, which due to scheduling constraints, resulted in having a limited number of test shifts.

Testing the REU behavior while sleeping was particularly challenging because the rover sheds the RCE and the REU switched power loads upon the sleep transition. With the RCE and FSW off, visibility into what the REU was doing was limited. However, 1553 bus logs provided insight into the REU registers and memory space while a change-of-state card in the MSTB produced time-tagged switch data.

The REU switched power is what powers its 1553 RT (remote terminal) – without it, external scripts cannot be sent to the REU to control and verify its performance. While the flight behavior has both REUs going to sleep at the same time, for testing purposes, scripts were designed to leave one REU in the awake state with the switched power for both REUs on, while the other one was the unit under test. This allowed complete visibility into the REU running Dream Mode and was necessary to verify a successful test. However, visibility came at a price of requiring a dual configuration testbed, which, depending on other testbed needs and re-work schedules, wasn't always possible.

6.2 Thermal Implementation

Curiosity's mission was comprised of multiple phases, each with a particular configuration and mode of operations. During the cruise phase, the RCE was always on and the thermal FSW module was responsible for constant thermal control of the cruise and descent stages. In contrast, the surface phase presented the constraints of pre-heating actuators before operational use and multiple shutdown and wakeup cycles. Separate thermal commands were used to handle these different behaviors, but both have the ability to turn on heaters. Also, the thermal FSW for both cruise and surface operations was initially designed to only be executed with the RCE on. With the introduction of Dream

Mode and the capability of warm-up heating while sleeping, yet another thermal command was created that provided a table-driven ability to control heaters. These multiple warm-up methods, all capable of turning on heaters with slightly different flavors, resulted in numerous problems during development and test and are responsible for operational complexity on the surface now.

Upon every shutdown, a table is executed that turns off any warm-up heaters that were accidentally left on by the operations team. A similar table is run when the rover wakes up which also turns off heaters to enforce a known boot configuration. Prior to Dream Mode, if this safety check was not in place, a heater could be left on all night and potentially damage the rover. With Dream Mode enabled and active, however, these table actions were discovered to interfere with table-driven warm-up heating during tests. To accommodate Dream Mode, the fix to this problem is to remove the heater actions from the table – with the consequence of not being able to use the old warm-up method without being exposed to the risk of accidentally leaving a heater on. While some of these issues were only discovered with integrated, end-to-end scenario tests, other problems could have been discovered earlier by designing the thermal FSW to better handle the different cruise, surface while awake, and surface while asleep behaviors.

6.3 Self-Heating and False Trips

Fault limits were hard-coded in the REU EEPROM for Dream Mode fault protection, and were intended to be high enough that they would never be reached except for failure cases. These fault limits were only expected to be exceeded when a warm-up heater switch was either left on or stuck on. If Dream Mode read a PRT above its fault limit, its fault protection code would turn on the RCE and let the FSW diagnose and respond to the problem. The FSW would attempt to turn off the heater, and if successful, only result in a warning. If it failed to do so due to a broken switch, the FSW would swap PAMs that would allow the heater to be turned off from the other side.

Nominal testing of drilling activities, however, revealed self-heating within the drill that caused the drill PRTs to exceed the defined thermal fault limits. Since the warm-up heater is not on while the drill is running, the FSW response is to issue a warning and re-assert the switch state off – and continue with the activity.

Dream Mode can only act upon the values of the PRT telemetry it collects and, unlike the FSW, is not capable of determining whether a fault limit was exceeded because of operational self-heating or a stuck-on switch. If Dream Mode was enabled and the rover shut down immediately following a too-hot drilling activity, Dream Mode would wake up the rover thinking there was a switch fault and the FSW would run a fault response – terminating sequences and safing the rover. Worse, depending on how long it would take for the drill to cool down, without disabling Dream Mode as part of the response, any subsequent shutdowns would also wake the rover up.

This deficiency in fault diagnosis resulted in creating flight rules that constrained the usage of Dream Mode around activities that may result in self-heating. Flight rules that are dependent on time make it difficult for the operations team to plan and schedule activities. A fix to avoid the self-heating and false trip problem was developed for the next software build, but the real lesson learned is to understand all of the circumstances involved with nominal and off-nominal scenarios, and then design the appropriate behavior to recognize and manage them all.

7 Acknowledgments

The REU FPGA design work by Jason Zheng made the Dream Mode capability possible. The Dream Mode algorithms and the integrated wakeup and shutdown behavior of Curiosity were developed by Tracy Neilson. Joe Stehly and Dhack Muthulingam tested the thermal FSW.

The rover design described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. © 2013. All rights reserved.