Can SeeStars Talk and Sleep? Developing Serial Communication and Power Management on SeeStar III Sensor Module

Beverly Abadines, California State University San Bernardino

Mentors: Thom Maughan, Chad Kecy

Summer 2017

Keywords: SeeStar, Imaging System, Low-Cost, Open-Source, Autonomous

ABSTRACT

Although autonomous vehicles provide a mechanism and relief from collecting extensive samples, they do face the price of being costly and cumbersome to deploy, limiting their numbers *and* accessibility to only a fraction of the marine science community. This directs the importance of an inexpensive, lightweight, open-source autonomous monitoring system, which can promote the convenience of having imaging systems available to a larger part of the community and public. This paper discusses the newest development of the low-cost, SeeStar project, which includes the serial communication protocol and power management on the Feather M0 found on SeeStar III's sensor module. The Feather M0 parsed and decoded commands from the Camera module and executed either Take Sample, Read Last, or Invalid request. On sleep mode, the Feather M0 operated at 3.3 mW and 99 mW during its normal operating mode, conserving 30 times as much energy when it is asleep than if left awake. Future contributions should implement peripheral sensors, discern any possible delays between sample recordings, and expand the protocol to include sensor accessor functions.

INTRODUCTION

The ocean houses myriads of organisms and environments, which through data collection in the form of images, pH, salinity, and other measurements has led to the discovery of scientific and social improvements (Hou, H.C. Bonser, & Jeronimidis, 2011 and Ruvinov & Cohen, 2016). The ocean, however, reaches over 11,000 m. in depth and composes 71% of the world, making the process of gathering copious and useful information throughout the majority of marine ecosystems nearly impossible when executed manually (noaa.gov, 2015 and Perlman, 2015). Consequently, autonomous devices equipped with scientific instruments, such as Autonomous Underwater Vehicles (AUV), Remotely Operated Vehicles (ROV), and Wave Gliders, that can independently monitor and accumulate information were developed, but although these vehicles provide a mechanism and relief from collecting extensive samples, they do face the price of being costly and cumbersome to deploy, limiting their numbers *and* accessibility to only a fraction of the marine science community. This directs the importance of an inexpensive, lightweight, open-source autonomous monitoring system, which can promote the convenience of having imaging systems available to a larger part of the community and public. Such a system is being developed by Monterey Bay Aquarium Research Institute (MBARI) and is known as SeeStar (mbari.org).

SEESTAR

SeeStar is an open-source, low-cost subsea imaging system used to autonomously capture images or videos during its months long deployment. It is composed of off-the-shelf materials making it cost-effective, while its small and modular design provide an adaptability and ease to attach it on moorings, tripods, or aquatic vehicles, and alleviate the difficulty of deployments that other independent monitoring systems encounter (Fulton-Bennett, 2015). Throughout its development, SeeStar has been used to survey fish found in marine protected areas at Monterey Bay, help AUV that are unable to capture images of translucent organisms detect zooplanktons, and study the benthic ecology at Antarctica (Cazenave, Kecy, Risi, & Haddock, 2014).

Unlike its predecessors, SeeStar III is designed to be assembled with peripheral instruments and collect sensor readings along with its ability to record images. This paper discusses the development of the SeeStar project, which includes the serial communication protocol and power management on the Feather M0 found on SeeStar III's sensor module.

MATERIALS AND METHODS

SEESTAR III : SERIAL COMMUNICATION AND POWER MANAGEMENT DESIGN

SeeStar III comprises a camera module and sensor module as shown in Figure 1.

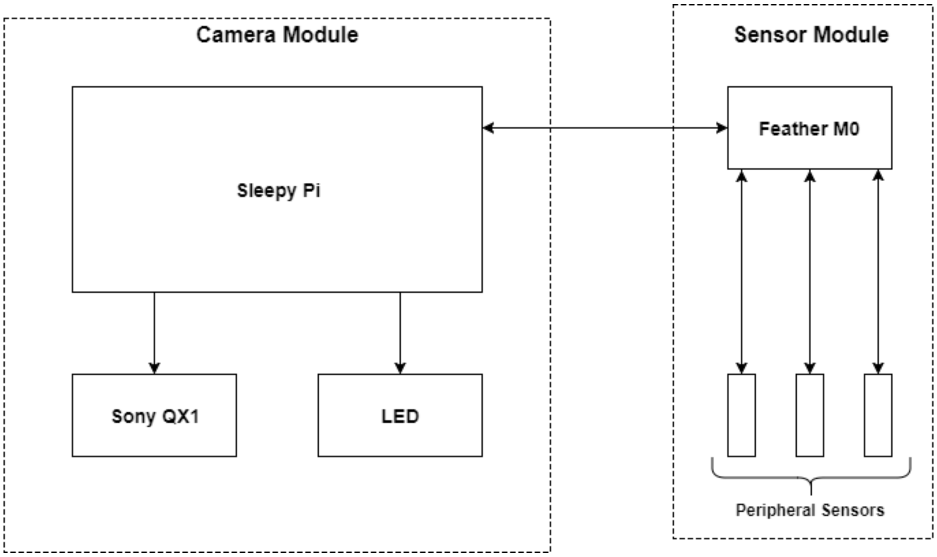

Figure 1. Block Diagram of SeeStar III: The Camera Module includes a camera and LED, which are operated by Sleepy Pi. The Sensor Module's potential peripheral sensors (O^2, pH, etc.) are controlled by the Feather M0.

The basic communication protocol between the two modules is to have the Camera Module wake up the Sensor Module from its default sleep state and have it ready to process commands, where the two commands are either to record samples taken by all the sensors onto the Sensor Module's SD card or to read the last log on file back to the Camera Module. The importance of the sleep state is to reduce the power consumption by only activating the Sensor Module once it is requested.

Command	Description
$1TSYYMMDDHHMMSSc	Take Sample: Record sensor sample on SD card
$1RLXXXXXXXXXXXXc	Read Last: Return last log on file

Figure 2. Each command is 17 bytes long with the last byte serving as a means for data integrity. YYMMDDHHMMSS is the time stamp relevant to the Camera module's real time clock and represents the time a request was made. It is recorded onto the log and stands for 2 byte representations of the year, month, day, hour, minute, and second. Read Last does not format with respect to time, so X's are used to fill the buffer.

PIN MUXING ON FEATHER M0 FOR SERIAL COMMUNICATION

One of the motivations for using the Feather M0 was its pin muxing capability which enabled the navigation of hardware through software configurations and allowed a maneuverability of serial connections and protocols to be mapped across different pins. The Feather M0 had serial communication or SERCOM ports ranging from SERCOM0 to SERCOM5 that can be mixed and matched to be SPI, I^2C, or UART, meaning that there are potentially 6-combination-3 possibilities for serial instantiations. The Sensor Module for this project only applied UART ports.

PRODUCING SENSOR READINGS

One of the SERCOM ports was used to engender data for the other sensor serial lines to read. Each time a Take Sample command was called the designated serial would write and then increment an unsigned 8-bit integer, extending from 0 to 255.

MEASURING CURRENT TO MONITOR POWER MANAGEMENT

To measure the current drawn, a 3.7 V battery was cut in half to separate the port from the battery as shown below.

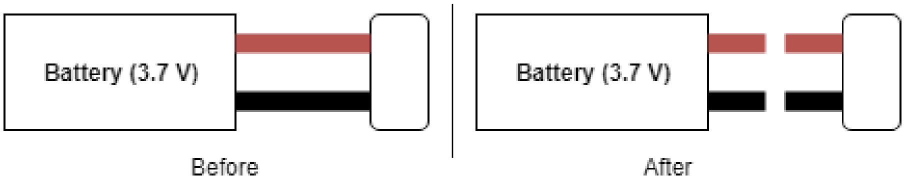

Figure 3. The battery and port are separated.

This provided wires for a resistor to be inserted between the Feather M0 and a DC supply running at 3.3 V. With the resistor in place, a multimeter measured the voltage across the resistor, and the current drawn was calculated using Ohm's Law, where current equaled the voltage across the resistor divided by its resistance. The power consumed was simply the voltage from the DC Supply multiplied by the calculated current.

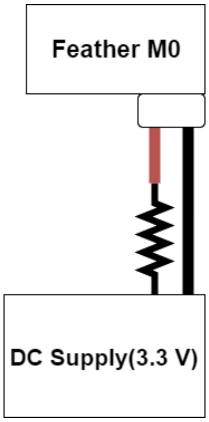

Figure 4. A resistor is inserted in series with the DC supply and Feather M0.

RESULTS

UART SERIALS

Only 4 of the 6 serial ports were available for the SeeStar III system given that SERCOM4 was used as a SPI to communicate to the SD card and SERCOM2 conflicted with the Master-In-Slave-Out (MISO) connection, which left only SERCOM0, SERCOM1, SERCOM3, and SERCOM5. SERCOM5 served as the main port between the two modules, SERCOM1 wrote out the dummy data, and SERCOM0 and SERCOM3 acted as the sensor ports, which collected readings. In practice, this resulted with 3 ports available for sensor peripherals and one port to serve as the main communication line between the Camera Module and Sensor Module.

SERIAL COMMUNICATION

The Sensor Module waited to receive a character delimiter to notify when to append its buffer. Once the buffer was full, it parsed and decoded commands from the Camera Module and executed either Take Sample, Read Last, or Invalid request as indicated from Figure 2.

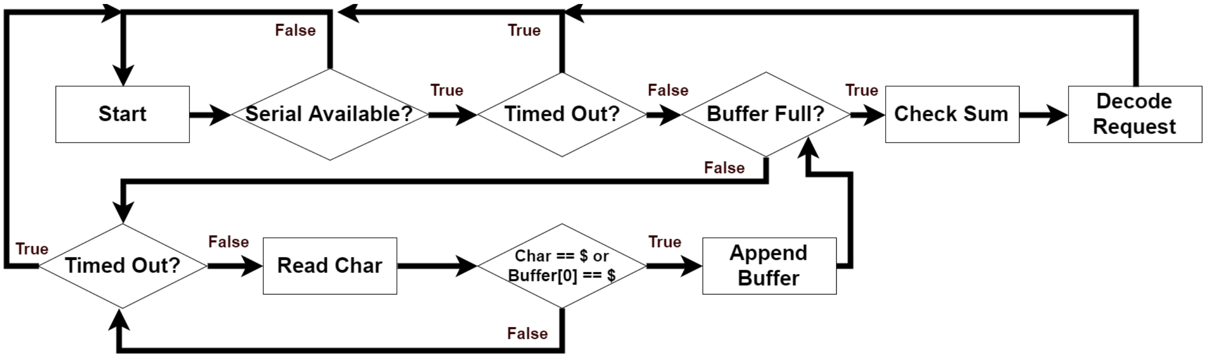

Figure 5. The serial communication protocol between the Sensor Module and Camera Module includes, timed out functions, a check sum function and a decode request function.

The timed out function compared the time that the Sensor Module was triggered to wake up with an arbitrarily set idle time. When the difference between the current time on the Feather M0's real time clock and the triggered time exceeded the idle time, the Sensor Module returned to sleep. This function's purpose was to remove any possible hang times and prohibit the module from drawing more current when no requests from the Camera Module were made.

The check sum function summed the 12 bytes of the time stamp and verified if the last byte of the command matched the calculated sum. The sensor module returned "ACK" if the two values matched or if the last byte was a don't care character, but "NACK" otherwise. The check sum served to confirm and indicate whether the data transmission from the Camera Module to the Sensor Module was uninterrupted and uncorrupted.

POWER MANAGEMENT

On boot up, the Feather M0 of the Sensor module disconnected its serial ports before it entered its power conservation mode, sleep. Once the Camera Module triggered the Feather M0 to wake up by sending a byte to one of the digital pins, the Sensor Module entered its normal operating mode and attached its serial ports to prepare for the serial communication protocol. On sleep mode, the Feather M0 operated at 3.3 mW and 99 mW during its normal operating mode, conserving 30 times as much energy asleep than if left awake.

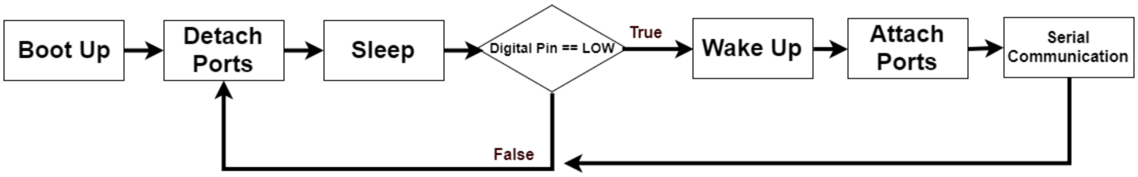

Figure 6. This is the design flow for the power management on the Feather M0 of the Sensor Module.

DISCUSSION

UART SERIALS

SERCOM4 was designated as the SPI line for the SD card, which could not be helped nor limited the availability of serial ports. SERCOM2, however, could have been available except that of its existing pins, only two were tangible hardware lines. From those two lines, one was a MISO connection, which could not be configured to operate for UART. It is worth considering to split the 3 sensor ports into channels and branch out lines and multiply the number of sensors.

SERIAL COMMUNICATION

Although the protocol provided communication between the two modules, a state machine could have been implemented to improve the system’s efficiency. For example, it may be used to ensure that once an Invalid request is decoded, the Sensor module immediately returns to its sleep state and avoids performing a check sum.

Additionally, Read Last could have included a time stamp such as the one for Take Sample in order to provide a value that can be used to verify the data integrity of the command. Read Last also could have returned the most recent data block rather than just the last individual line of the sensor readings. Furthermore, to conserve energy, the Feather M0 should access a local variable that stores the same values written to the SD card when Take Sample is called rather than opening and reading a file every time the Camera Module asks for Return Last.

POWER MANAGEMENT

The recorded power consumed was respect to the Feather M0 and its connection to an RS232 adapter without any sensor peripherals. The current drawn will possibly be higher once sensors are integrated.

CONCLUSIONS/RECOMMENDATIONS

SeeStar explores the possibility of serving as an open-source, low-cost, and modular monitoring system that can be used to record images and aquatic sensor readings between its Camera and Sensor modules. This paper presents a serial communication protocol between the two modules, which coordinates the ability to send, read, and execute commands, such as Take Sample or Read Last, as well as being energy efficient through the use a default sleep state. Future contributions should implement peripheral sensors, discern any possible delays between sample recordings, and expand the protocol to include sensor accessor functions.

Continuation of the open-source code and documentation can be found at https://bitbucket.org/mbari/seestar and at https://github.com/BeverlyAb.

ACKNOWLEDGEMENTS

Thank you to my main project mentors, Thom Maughan, Chad Kecy, Danelle Cline, and Jose Rosal. Also, thank you George Matsumoto and Linda Kuhnz for coordinating this opportunity. Even though I learned and applied so much work into this project, I do not think I can ever learn enough from my mentors.

Thank you MBARI, the David and Lucile Packard Foundation, and all the engineers, technicians, and scientists. MBARI's faculty are as brilliant as they are humble and down to Earth.

Lastly, thank you Rascal.

References:

Bitbucket. (n.d.). Retrieved June & july, 2017, from https://bitbucket.org/mbari/seestar/src

Cazenave, F., Kecy, C., Risi, M., & Haddock, S. H. (2014). SeeStar: A low-cost, modular and open-source camera system for subsea observations. Oceans -St. John's, 2014. doi:https://doi.org/10.1109/OCEANS.2014.7003077

Fulton-Bennett, K. (2015, January 21). New SeeStar camera system allows researchers to monitor the depths without sinking the budget. Retrieved June & july, 2017, from http://www.mbari.org/new-seestar-camera-system-allows-researchers-to-monitor-the-depths-without-sinking-the-budget/

Hou, J., Bonser, R. H., & Jeronimidis, G. (2011). Design of a Biomimetic Skin for an Octopus-Inspired Robot – Part I: Characterising Octopus Skin. Journal of Bionic Engineering, 8(3), 288-296. doi:10.1016/s1672-6529(11)60032-2

Perlman, U. H. (2015). How much water is there on, in, and above the Earth? Retrieved June & july, 2017, from https://water.usgs.gov/edu/earthhowmuch.html

Ruvinov, E., & Cohen, S. (2016). Alginate biomaterial for the treatment of myocardial infarction: Progress, translational strategies, and clinical outlook: From ocean algae to patient bedside. Advanced Drug Delivery Reviews, 96, 54-76. doi:https://doi.org/10.1016/j.addr.2015.04.021

SeeStar Imaging System. (2016, January 07). Retrieved June & july, 2017, from http://www.mbari.org/technology/emerging-current-tools/seestar-imaging-system/

US Department of Commerce, National Oceanic and Atmospheric Administration. (2015, June 17). How big is the Pacific Ocean? Retrieved June & july, 2017, from http://oceanexplorer.noaa.gov/facts/pacific-size.html