Imperial College London

IMPERIAL COLLEGE LONDON

SECOND YEAR DESIGN PROJECT

ELEC50003/ELEC50008

The MARS Rover

Group 1

Georgio Chaimali
Dimitrios Georgakopoulos
Edvard J. Skaarberg Holen
Hyunjoon Jeon
Josiah Mendes
Raghav Viswakumar

Word Count: XXXX Words
June 7, 2021

Contents

1 Overview

2 Systems

2.1 Control

2.2 Comms

2.3 Vision

Abstract

The purpose of the Vision module is threefold: 1. Capture data from camera module; 2. Detect objects of interest within the current view and send their location to the Control module; and 3. Send image data to Control for streaming to Command.

2.3.1 Hardware Organisation

The Vision module comprises of two main hardware elements: the Terasic DE10-Lite, a cost-effective Intel MAX 10 based FPGA board [1] and the Terasic D8M-GPIO camera package [2] that interfaces with the FPGA through the onboard GPIO connectors.

These hardware choices were made by the project organisers, but are also sufficient and capable of carrying out the tasks at hand. As the FPGA's hardware is configurable, it is more flexible than other embedded systems that are limited to a general purpose processor,and is also able to handle both streaming and processing of high resolution images without significant compromises on framerate or data speed through the use of concurrent operations and dedicated blocks for signal processing applications like multiplication. This particular FPGA is also equipped with a 4-bit VGA output which is useful for debugging object detection live, and also has a connector for an Arduino Uno R3 shield, [1] which can be used to interface with the ESP32 used for control.

In order to perform general purpose operations like to configure camera settings and to provide a debugging interface, a Nios II soft core was instantiated on the FPGA. Alternatively, to implement a more advanced image processing algorithm or to reduce other hardware components in the system like the multiple Arduinos, a FPGA with a hard core, known as a FPGA System-On-Chip (FPGA SoC) [3] could be used, which would provide both the advantages of having reconfigurable hardware and a more capable general purpose processor.

2.3.2 Image Capture Processing Stream

The image capture and buffering is based on a starter project provided by Terasic Inc for the D8M Camera module that was modified by Ed Stott [4].

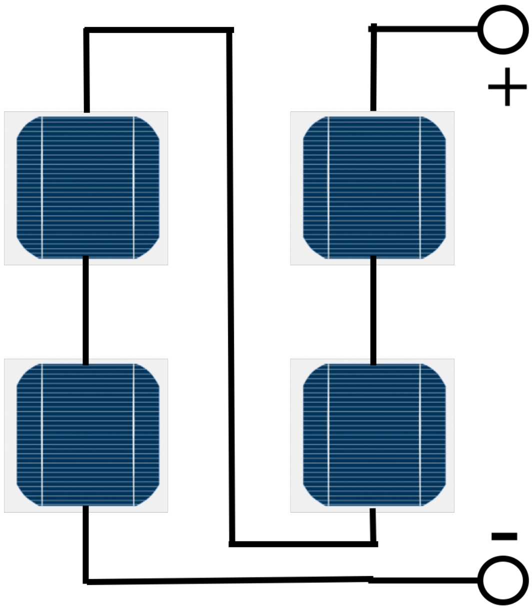
2.4 Drive

2.5 Energy

Abstract

2.5.1 Characterising Components

2.5.2 PV Array Configuration



2.5.3 Battery Configuration

2.5.4 SMPS Configuration

2.5.5 Maximum Power Point Tracking

2.5.6 Charging Algorithm

2.5.7 Discharging Algorithm

2.5.8 Safety Mechanisms

2.5.9 State of Charge

2.5.10 State of Health

2.5.11 Communicating with Other Modules

Though it is not necessary to fully integrate the energy module with the rest of the rover, other submodules, specifically command, needs access data such as the battery SOH and SOC. For communicating with other modules the Arduino shield has a set of UART ports. However, as group members were not in the same location it was not possible to physically connect the energy module to the rover, which is necessary to use UART. As such, an alternative approach was employed. First the Arduino was connected to a computer via USB. On the computer a Python script was run [8]. At the start the Python script establishes a connection to a server created by running a similar script on the command module [9]. After a connection has been established the Python script starts reading the serial data coming from the Arduino and transmits it using TCP to the command module. Each message coming from the Arduino is in CSV form where the first entry is the message ID, which allows the command script to decode what type of data is being sent.

2.6 Integration

2.6.1 Integrating the Energy Module

3 Evaluation and Conclusion

4 Project Management and Organisation

As this project was carried out remotely with contributors located in different countries, it was important to have a good framework for communication and management.

The main tool used for communications and management was Git + GitHub. As the codebase was incredibly complex, involving many different libraries and with each submodule being capable as a standalone project, it was vital to have a version control system in place. Being able to keep a history of commits and changes made to a project was useful, especially when trying to track down the origin of a bug and what caused it.

The team also made use of GitHub Issues to track progress and accountability in the initial design phase. A thread was opened for each submodule to show what the lead for that submodule had been doing and potential avenues of achieving their goals. This was beneficial both for the leads to keep track of their research, but also allowed other members to contribute to other submodules by adding comments and voicing their thoughts. GitHub Issues were also linked directly to commits in the codebase to allow for a more in-depth explanation and reasoning with context for a commit than what is allowed in the commit message area.

5 Intellectual Property

References

[1] Terasic Inc. “DE10-Lite Board.” (), [Online]. Available: `https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English%5C&No=1021`.

[2] ——, “D8M GPIO - 8 Mega Pixel Digital Camera Package with GPIO interface.” (), [Online]. Available: `https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English%5C&No=1011`.

[3] Intel Corporation. “Intel® FPGAs & SoC FPGAs.” (), [Online]. Available: `https://www.intel.co.uk/content/www/uk/en/products/details/fpga.html`.

[4] edstott, *EE2Rover*, `https://github.com/edstott/EEE2Rover`, 2021.

[5] A. Einstein, “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies],” *Annalen der Physik*, vol. 322, no. 10, pp. 891–921, 1905. DOI: `http://dx.doi.org/10.1002/andp.19053221004`.

Figure 1: Sample Figure

Sample Reference[5]

Appendices

Some Appendix The contents...