Project Report  
  
**DcPi  
Underwater Camera System**

****

*Prepared by*

Arthur Bondar   
May 2, 2018

Contents

[**Overview** 3](#_Toc513151766)

[**Operation** 3](#_Toc513151767)

[**Components** 4](#_Toc513151768)

[**Interface** 5](#_Toc513151769)

[**Output** 6](#_Toc513151770)

[**Duration** 7](#_Toc513151771)

[**Bill of Material** 8](#_Toc513151772)

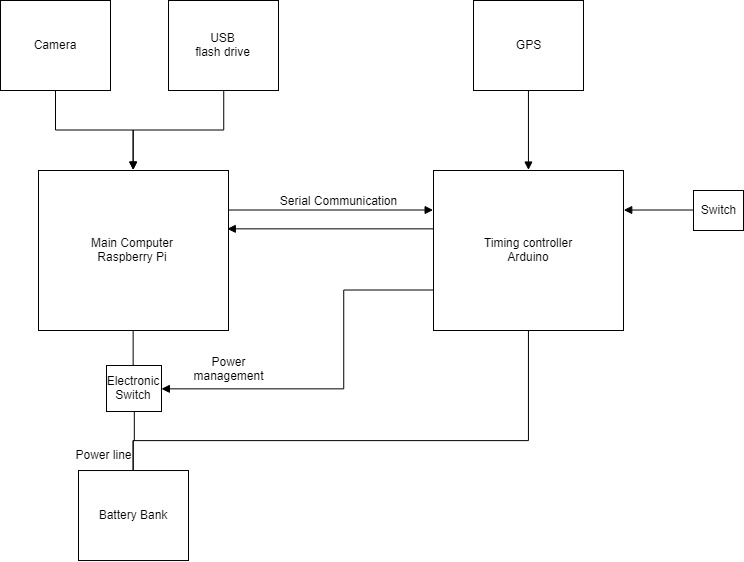
# **Overview**

Under Water Camera System is designed to collect videos and sensor data and video from Leatherback sea turtles for Marine Biology research. The devices includes multiple microcontrollers and is currently capable of video and GPS location data storage. Since a dedicated microcomputer is used as a main controller, more data collection devices i.e. sensors and transducers, can be added easily.

System include rechargeable lithium ion batteries and is design for a multiple-day data collection mission. Using user defined video time-lapse, camera can record videos during the day and remain in a shutdown state to conserve battery during the night. Future development can include a timed release valve system, which can easily be integrated with the current system. This document intends to describe system operation from technical point of view for potential future development. Instructions on deployment will be described in a separate document.

# **Operation**

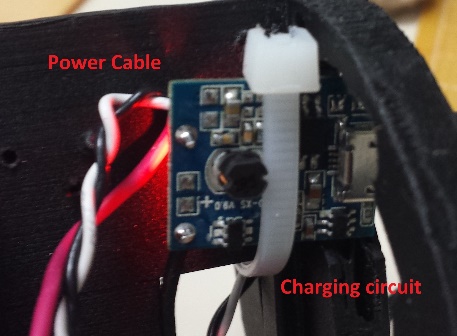
User sets time-lapse video timing in the main computer (Raspberry Pi) through a USB. Both time values communicated into a smaller micro-controller (Arduino). This controller communicated with a GPS module, which mounted on the same board, to determine accurate time value. When current time is out of recording time range, Timing controller send a sleep command to the main Pi computer. When this command received, video recording is interrupted and the computer shuts down. After shutting down, the power line to the main computer is interrupted to conserve power. Once Timing controller verifies that current time is within the time for video, power to main computer is restored and video recording begins. The cycle continues until it is interrupted by the user then deployment is finished or when battery is drained out of energy.



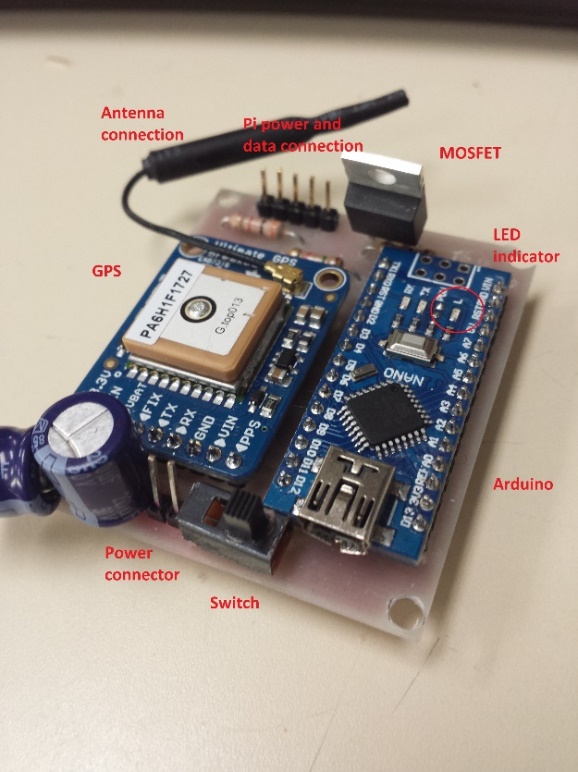
# **Components**

The system can be separated into three main modules: battery and charger, timing controller and GPS (Blue boards), camera and main computer (Green board).

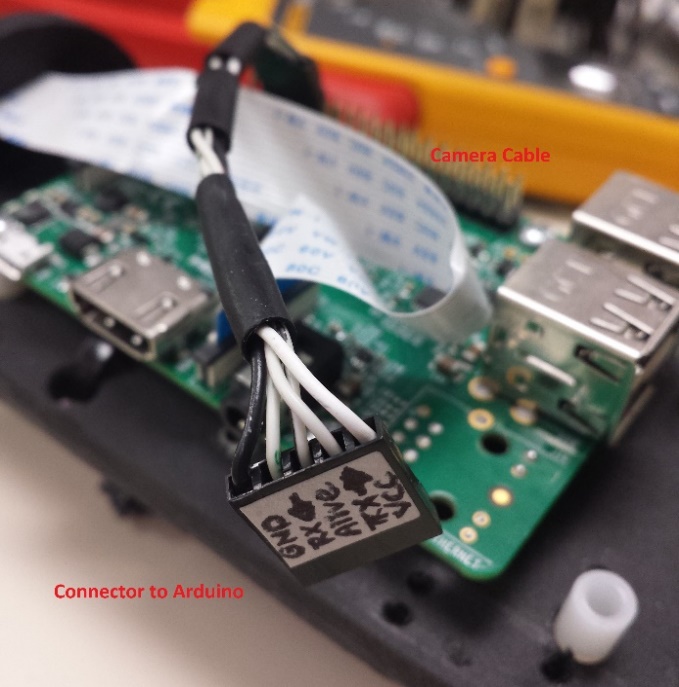
Battery and charging circuit

This circuit provides stored energy for the system. Battery charging circuit is used to recharge the cells and to convert low battery voltage (3.6-4.2 volts) into a steady voltage (5 volts) needed for the circuit operation. Battery consists of 8 lithium ion 18650 cells. Each cell has approximate capacity of 3.35Ah providing a total capacity of 26.7Ah. All cells are arrange in parallel and should be examined carefully before and after each deployment. Cell voltage should not be below 3 and exceed 4.2 volts. Depleted battery will have a voltage of approximated 3.6v, while a charged pack will have a voltage of 4.1 volts. Charging circuit have LED indicator. Flashing red light indicates charging, solid red – fully charged. Solid blue light indicates that current is being drawn from the batteries – during operation.   
**Note**: although present charging circuit have two ports, the device cannot be power while the batteries are being charging.

Timing and GPS

Timing printed circuit board consist of a GPS (Ultimate GPS from adafruit), Microcontroller (Arduino), electronic and mechanical switches. This board has small current consumption and stays powered through the whole deployment. It is used to monitor current time and schedule start and end of video recording. It uses a mechanical switch to indicated beginning and end of deployment, as well as electronic switch (MOSFET) to regulate the power to the camera board. The controller is able to receive current time and location from a GPS module and commutate it to the main camera computer (Rasbperry Pi). A 5 pin connected is used to connect data and power between this board and the Raspberry Pi. Recording loop starts when the switch is turned into ON position. LED light labeled ‘L’ on the board is used to provide indication when a recording loop is started.

Camera and main computer

Main computer used in this system is Raspberry pi 3 model B together with a Pi camera version 2. This computer is responsible for video recording and storage. Its main interface is a USB flash drive, which contains deployment timing – **mission.txt**, as well as the final video recordings and GPS data. Initially, when deployment start this computer reads the USB and transmits data over to the Timing computer. When data is transmitter, the computer requests current time and begins video recording. In parallel, this computer listens for incoming sleep commands from timing controller to indicated end of video cycle. Finally, this controller is programs to request GPS data from Timings controller at a 2 minute interval. This data is time stamped and recorded in Excel-compatible file.

# **Interface**

Main user interface is done through a USB flash drive. USB drive should be formatted as **exFAT** in order to accommodate large file sizes. Inside the USB drive, a file – **mission.txt** can be used to set recording time frame. This time determined the camera recording activity, it is made adjustable for the user to make daylight adjustments or perform a longer deployment with short video intervals. Format of the mission file is as follows:

|  |  |
| --- | --- |
|  | {  "start": "13:00",  "end": "13:05",  "duration": "999",  "width": "1640",  "height": "922",  "frames": "25",  "rotate": "180",  "gps": "2"  } |
|  |  |

Where start and end are the end times in 24 hour format, and duration specifies full mission duration in minutes (currently not implemented with release valves). In the example above, the camera will only record for 5 minutes each day for a whole duration of 999 minutes. In present implementation, duration field can be populated with any number. If mission file was accidently deleted, it needs to recreated with the same date and data inside.

**Note:** It is important to keep the format of the file as it appears above, including all brackets and braces. Each parameter pair is separated by comma, except for the last one.

Timing parameters – **start** and **end** times affect video duration, since camera board has the largest energy consumption, those times can be used to shorten the video interval and spread the recording across more days of deployment.

A switch on the Timing board can be used to start deployment and recording loop. A small LED on the Arduino board can provide indication whether the board read switch position successfully. Final indication of system performance is LED’s on Raspberry Pi camera board. When Pi board is powered a recording begins and LED indicators light up.  
**Note:**  If switch was turned on and mission started before “start” time in the file. Pi will boot up and start communication with timing board. When timing controller received the start time, it will check current time and send sleep command which will shut down the main Pi system until the right start time.

# **Output**

DcPi uses a USB flash drive to record collected data. The USB must be formatted in **exFAT** to support large file sizes for the video**.** Upon successful deployment, system produces three types of files – Video recording and CSV data file and a log file. Video and GPS files will be created for every day during the deployment and have a proper timestamp.

* **Video recording** – Video recording will be named with a timestamp in the following form: YYYY-MM-DD\_hh\_mm\_ss. Video format is **.h264** and need to be converted into mp4 or other formats. There are free converters available for Windows, MAC and Linux.
* **GPS data** – data is written in a spreadsheet compatible file – CSV. Name of the file is a timestamp with the following format YYYY-MM-DD, parameter **“gps”** can be used to set GPS recording interval in minutes.
* **Troubleshooting log** – log file called **turtle.log** has information about system performance for all previous deployments. It is useful for developers to troubleshoot the software and find faults.

All files except for **mission.txt** can be deleted from the USB drive after each deployment.

# **Duration**

Two main factors are taken into account in order to calculate mission duration – battery life and disk usage.

Battery life

Largest power consumption is the camera board – Rasbperry Pi. Since On-time of this board can be adjusted through the mission file, full deployment interval can be extended.  
The following were used to calculate battery life:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Power Consumption 24 Hours** | | | | |
| **Device** | **On Time (hrs.)** | **Current (mA)** | **Energy (Ah)** | **Energy (Wh)** |
| RPI + Camera | 10 | 420 | 4.2 | 21 |
| Arduino | 24 | 80 | 1.92 | 9.6 |
| TOTAL: |  |  | 6.12 |  |

Using current 26.7Ah battery, the following times have been estimates:

|  |  |  |  |
| --- | --- | --- | --- |
| **Battery Matching** | | | |
| **On Time (hrs.)** | **Capacity (ah)** | **Duration (hours)** | **Duration (days)** |
| 6 | 26.7 | 144 | 6.0 |
| 8 | 26.7 | 121 | 5.1 |
| 10 | 26.7 | 105 | 4.4 |
| 12 | 26.7 | 92 | 3.8 |

Disk space

The system currently uses a 256GB USB flash drive for video storage. Video file size is affected by resolution, frames per second and timing values. Both can be adjusted in mission parameters file.

The following is a quick calculation of recording file sizes using 12 hour recording per day.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Video Storage Calculation** | | | | |
| **Resolution** | **FPS** | **Duration (hrs)** | **per 1 sec (mb)** | **Total (gb)** |
| 1920x1080 | 30 | 12 | 1.5 | 64.8 |
| 1640x922 | 25 | 12 | 0.61 | 26.35 |

|  |  |  |
| --- | --- | --- |
| **Storage Matching** | | |
| **Disk Size** | **Resolution** | **Duration Days** |
| 256 | 1640x922 | 9.71 |
| 1920x1080 | 3.95 |

# **Bill of Material**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bill Of Material Electrical** | | | | |
| **Component** | **Quantity** | **Cost** | **Total Price** | **Link** |
| Raspberry Pi 3 Model B | 1 | CA$99.00 | CA$99.00 | [Pi 3 Model B](https://www.amazon.ca/CanaKit-Raspberry-Complete-Starter-Kit/dp/B01CCF6V3A/ref=sr_1_2?ie=UTF8&qid=1525398379&sr=8-2&keywords=raspberry+pi+3) |
| Pi Camera V2 | 1 | CA$40.00 | CA$40.00 | [Pi Camera V2](https://www.amazon.ca/Smraza-Raspberry-Camera-Module-Supports/dp/B076KCZRDS/ref=sr_1_1_sspa?ie=UTF8&qid=1518824426&sr=8-1-spons&keywords=raspberry+pi+camera&psc=1) |
| Adafruit GPS | 1 | CA$60.00 | CA$60.00 | [Ultimate GPS](https://www.adafruit.com/product/746) |
| Arduino Pro Mini | 2 | CA$11.00 | CA$22.00 | [Arduino Pro Mini](https://www.amazon.ca/Arduino-Pro-Mini-328-16MHz/dp/B0089TPH3O/ref=sr_1_2?ie=UTF8&qid=1518963555&sr=8-2&keywords=arduino+pro+mini&dpID=51XA7OJo1JL&preST=_SY300_QL70_&dpSrc=srch) |
| USB flash disk 256gb | 1 | CA$100.00 | CA$100.00 | [SanDisk 3.0](https://www.amazon.ca/SanDisk-Ultra-Flair-256GB-SDCZ73-256G-G46/dp/B06XG9XP49/ref=cm_cr_arp_d_product_top?ie=UTF8) |
| Battery Bank | 1 | CA$50.00 | CA$50.00 | [26ah Battery Bank](https://www.amazon.ca/Battery-RAVPower-Charger-26800mAh-Portable/dp/B012S6IHQC/ref=sr_1_9?s=electronics&ie=UTF8&qid=1518824581&sr=1-9&keywords=battery+bank) |
| MOSFET IPB80P03P4L-04 | 1 | CA$5.00 | CA$5.00 | [DataSheet](https://www.infineon.com/dgdl/Infineon-I80P03P4L_04-DS-v01_01-en.pdf?fileId=db3a30431ddc9372011e07e95eb827d7) |
| Resistors - 10k, 1k, 2k | 5 | CA$1.00 | CA$5.00 |  |
| Capacitors 220uF | 2 | CA$1.00 | CA$2.00 |  |
| **TOTAL:** |  |  | CA$383.00 |  |
| **Alternatives** | | | | |
| USB Hard Drive 2TB | 1 | CA$90.00 | CA$90.00 | [Seagate 2TB Drive](https://www.amazon.ca/Seagate-Backup-Portable-External-STDR2000100/dp/B00FRHTSK4/ref=sr_1_3?ie=UTF8&qid=1519561265&sr=8-3&keywords=usb%2Bhard%2Bdrives&th=1) |