



Hardware Article

CoralCam: A flexible, low-cost ecological monitoring platform

Austin Greene^{a,b,*}, Zac Forsman^b, Robert J. Toonen^b, Megan J. Donahue^b^a University of Hawaii Manoa, United States^b Hawaii Institute of Marine Biology, United States

ARTICLE INFO

Article history:

Received 30 May 2019

Received in revised form 17 October 2019

Accepted 18 October 2019

Keywords:

Coral
Marine
Camera
Arduino
Ecology
Monitoring
Low-cost
Conservation
Management
Terrestrial
Movement
Habitat

ABSTRACT

The study and conservation of biological communities, such as coral reefs, frequently requires repeated surveys to measure the growth of organisms or the occurrence of ecological processes, such as recruitment, predation, competition, or mortality. In the case of coral reefs, processes influencing coral community structure occur on time scales of days (recruitment, predation), months (seasonal environmental stress), or years (competition for space). In both marine and terrestrial systems, observing the ecology of remote locations at fine temporal scales is made difficult by the high cost or complexity of resurveying the same location at high frequency. These restrictions have produced limited understanding of *in-situ* ecological processes which occur at fine temporal scales and influence community structure but are easily missed during infrequent surveys. We present a low-cost method for the conversion of consumer cameras into programmable time-lapse platforms, allowing scheduled daily video or photo capture in remote locations for extended time periods. Results of a 1-month deployment with twice-daily photo capture are presented. Total cost to construct and deploy CoralCam *in-situ* (up to 45 m) is approximately \$80 USD, providing a low-cost platform for fine scale data collection where these data are not otherwise logistically or financially possible.

© 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Specifications table

Hardware name	Camera (generic “action camera”)
Subject area	Biological Sciences
Hardware type	Field measurements and sensors
Open Source License	Creative Commons Attribution-ShareAlike license
Cost of Hardware	<\$80.00 USD
Source File Repository	https://osf.io/dum2h

1. Hardware in context

The study of ecological processes and human impacts in marine or terrestrial habitats often requires frequent survey by human observers or the extensive use of camera traps [1,2]. For example, assessment of coral or fish recruitment is often used as a baseline indicator of habitat resilience on coral reefs [3]. *In-situ* observations are frequently used to establish

* Corresponding author.

E-mail address: AustinLG@Hawaii.edu (A. Greene).

the presence or severity of coral bleaching, seasonal environmental stress, competition, animal activity, or human impacts [4–8]. However, ecological processes occurring at daily timescales or over longer time periods are frequently missed during comparisons of monthly, quarterly, or bi-annual surveys. In marine settings this is due primarily to the high risk and cost of deploying human observers to field sites; further complicated at deeper sites as depth limits bottom time on SCUBA or requires decompression. Similar problems arise in the study of organisms which are cryptic, remote, or may avoid contact with humans [9–13]. In the case of coral ecology, scarcity of *in-situ* daily observations has resulted in poor understanding of the rapid ecological processes affecting coral recruitment and persistence, such as recruit competition, colony fusion, or high-mortality events such as predation or bleaching [3–5]. In addition, despite widespread bleaching and disease stress on global coral reefs, a lack of daily *in-situ* observations has left disease and bleaching progression rates poorly described in the literature [4,5,14]. The need for such data is especially apparent in the face of current high-mortality coral tissue loss disease originating in Florida and spreading across the Caribbean [14]. Accessible, low-cost methods to reliably collect daily-scale visual data are critical to improved conservation and management in both marine and terrestrial settings.

The hardware described herein comprises a low-cost microcontroller (Arduino Pro Mini 3.3 V 8 MHz) interfaced with an inexpensive “action camera” style of consumer camera using off-the-shelf components to allow programmable capture of photos or video beyond functions normally available in the camera [15]. To our knowledge, there is only one commercial product designed to accomplish a similar result which is sold by CamDo Solutions (Vancouver, British Columbia) under the Blink product line [16]. The CamDo Blink product line (\$289–\$415 USD at the time of writing) includes a microcontroller-integrated circuit board which is plugged into the back of a GoPro brand camera to enable low-power, WiFi enabled time lapse functionality. All CamDo Blink models interface with the camera by exploiting outward-facing connectors on the GoPro camera itself and are not suitable for use on other, more-affordable action camera brands. At the time of writing, the total cost of a CamDo Blink and GoPro HERO4 camera is \$589.00 USD (or up to \$715.00 USD for a GoPro HERO5 with CamDo UpBlink). While these solutions are undoubtedly convenient, they are not financially conducive to large sample sizes needed for ecological studies or high rates of instrument loss in the field. An open-source camera for deployment in intertidal habitats has been designed by Dr. Luke Miller at San Diego State University, California which allows low-resolution (2MP or 5MP) image capture with excellent battery life [17]. However, this method does not fit in inexpensive consumer camera housings for underwater deployment and requires advanced soldering of surface-mounted components to a printed circuit board (PCB) with the use of a solder reflow oven. Our design represents a middle-ground between commercially available solutions which are expensive but readily available, and other customized camera solutions which are inexpensive but require specialized equipment or significant time to construct. CoralCam is open-source, easy to build, low-cost, and may be used on a wide range of available cameras. CoralCam does not require permanent modification of the camera. Based around an open-source Arduino microcontroller, there is abundant opportunity for end-users to integrate additional instrumentation to meet research needs, such as readily available low-cost temperature or light sensors. Capacity to capture photo or video segments in addition to concurrent environmental data is especially valuable for ecological or conservation management studies where environmental parameters may condition biological responses. A real-time clock is included in the design to allow data collection at user-specified times and coordinated or simultaneous data collection across an unlimited number of independently deployed cameras.

By integrating a microcontroller into a standalone camera, our approach allows users to easily program the type of data collected (photo or video), duration, and frequency without the need to program more complex processes such as image processing or storage. As a low-cost waterproof camera platform, CoralCam has ample use cases in other research areas which may benefit from long-term, high-frequency image capture such as nest observation, monitoring of human impacts, seasonal phenology, or observation of organisms that are sensitive to the presence of human observers [2–7,12–14].

2. Hardware description

CoralCam is constructed from readily available, low-cost “off the shelf” components and requires no specialized expertise to assemble. Wiring may be done by hand following the included schematic (Fig. 2) or by ordering low-cost CoralCam printed circuit boards (PCBs) from a supplier of choice using the freely included PCB schematic. Using the included PCB limits soldering to user-friendly through-hole connections and a minimal number of wired connections – no surface mount components are used. CoralCams assembled using the provided PCB benefit from a smaller footprint, improved structural reliability, and significantly reduced build time at an overall cost increase of approximately \$1.00 USD. Unlike available commercial products, assembly of CoralCam does require non-permanent modification of the camera. Compared to commercial products, CoralCam represents a total cost per unit of less than \$80.00 USD allowing for high rates of instrument loss which are common to conservation monitoring or long-term ecological field studies. An Arduino Pro Mini 3.3 V microcontroller electronically controls camera power, mode, and shutter functions through the use of two transistors (BC547 NPN). The microcontroller is highly power efficient and is powered by a small external battery for redundancy. All microcontroller programming is accomplished through the user friendly and open-source Arduino Interactive Development Environment (IDE). A DS3231SN (alternatively DS3231M) real-time clock wakes the microcontroller from a low-power sleep state at a frequency determined by the user, at which point the microcontroller turns on the camera, adjusts the camera to the desired capture mode (photo or video), captures imagery, and stores these data to a microSD card in the camera. Multiple



Fig. 1. A fully assembled CoralCam unit consisting of the modified camera, CoralCam PCB assembly, and microcontroller battery fitted into a GoPro Hero 4 underwater housing with “extended back door”.

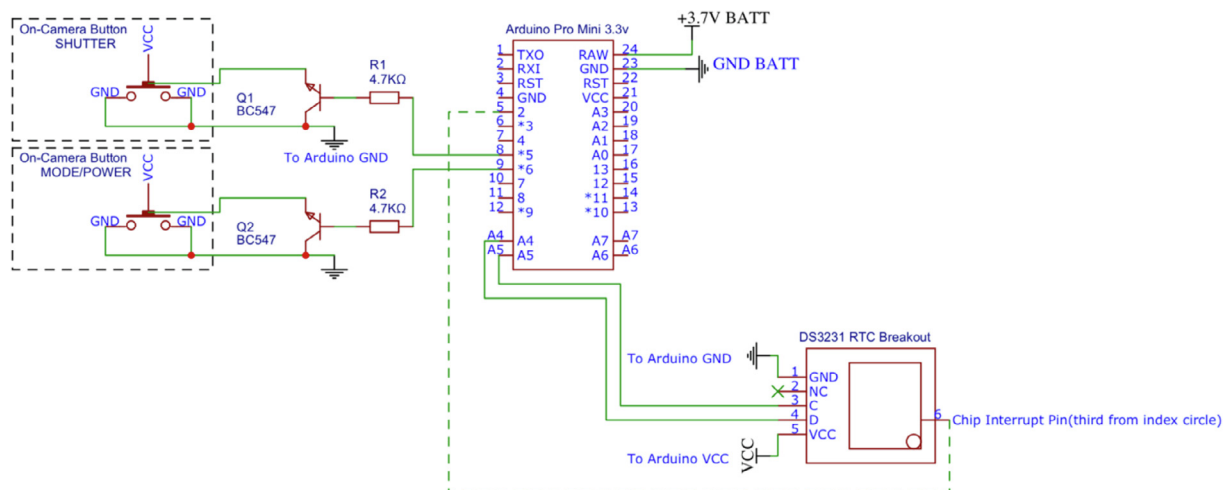


Fig. 2. CoralCam wiring diagram detailing electrical connections between the camera buttons, microcontroller, microcontroller power source, and real time clock.

daily captures are accomplished by the microcontroller sequentially reprogramming internal interrupt alarms on the real-time clock after each data collection period.

CoralCam deployment time is limited in order by 1) biofouling rates, 2) capacity or dissipation of the camera battery, and 3) capacity or dissipation of the microcontroller battery. Power consumption by the camera averages 260 mA when powered on and spikes to 320 mA during video or photo capture. Laboratory tests demonstrated that the camera battery allowed for 395 continuous 30-second cycles of powering on, photo capture, and powering off – or 79 similarly programmed cycles to capture 30-second video segments. Under real-world conditions with variable temperature and camera battery dissipation rates, two 30-second cycles (photo or video) per day provided the longest deployment time. Using the microcontroller to power off the camera between capture periods greatly extends camera battery life from a manufacturer rated runtime of 90 min to approximately 1 month of twice-daily 30-second periods of data collection. Hardware modification of the Arduino Pro Mini 3.3 V microcontroller reduce microcontroller power consumption to approximately 4.0 mA when active and 0.05 mA when in a low-power sleep state. CoralCam's ability to remain deployed for extended periods of time greatly reduces the cost of manual daily data collection. Monthly rotation or redeployment of CoralCam also provides a method for the low-cost, continuous monitoring of remote habitats or animal populations at daily scales – a capacity we believe to be largely unprecedented.

The camera, CoralCam PCB assembly, and microcontroller battery fit inside a consumer-grade underwater “action camera” housing (e.g., GoPro Hero 4 housing with extended back door) which allows for immediate deployment up to 45 m in marine habitats. The low profile of CoralCam also allows it to fit into larger housings with more batteries for longer deployments, specialized housings equipped with lighting, or housings designed to carry GoPro-style cameras for deep-sea deployment. These methods may be used to collect photos or video in a variety of other habitats where high-frequency data collection and minimal human impact is desired or logistically required. For validation of these methods we have designed and 3D-printed a coral recruitment monitoring platform (CRMP) consisting of a mount for CoralCam, a coral settlement tile, and a base plate to fix these components onto the substrate or dive weight. This platform is used to demonstrate the ability of CoralCam to collect daily photographs of a coral fragment and highlight small-scale habitat variability such as sediment stress.

Suggested uses for CoralCam include:

- Assessing ecological processes of corals, fish, or other marine organisms across environmental gradients.
- Measurement of disease progression, bleaching, or wound-healing rates on coral colonies
- Monitoring of animal populations or organisms which are sensitive to the presence of human observers (terrestrial or marine).
- Long-term, remote monitoring of habitat loss or occupancy at small spatial scales.

3. Design files

All design files are available and maintained at the Open Science Framework source file repository for this manuscript. Design files summary

Design file name	File type	Open source license	Location of the file
CoralCam assembly pre-deployment	Figure (PNG)	CC BY-SA	Available with article (Fig. 1)
CoralCam_Wiring_Diagram	Figure (PNG)	CC BY-SA	Available with article (Fig. 2)
CoralCamPCB_v1.2-Simple	.zip compressed Gerber file	CC BY-SA	Source File Repository
Coral recruitment monitoring platform assembly (CRMP)	Figure (PNG)	CC BY-SA	Available with article (Fig. 3)
CRMP_Base_Plate	STL	CC BY-SA	Source File Repository
CRMP_Settlement_Tile	STL	CC BY-SA	Source File Repository
CRMP_Base_Plate	STL	CC BY-SA	Source File Repository
CoralCam_TwiceDailyPhoto_ArduinoCode	.ino	CC BY-SA	Source File Repository
CoralCam_TwiceDailyVideo_ArduinoCode	.ino	CC BY-SA	Source File Repository

- *CoralCam assembly pre-deployment*: A photograph demonstrating components when housed inside a consumer-grade underwater housing prior to deployment.
- *CoralCam wiring assembly*: A photograph visually highlighting the interface of the Arduino microcontroller with the consumer camera via two transistors.
- *CoralCam_Wiring_Diagram*: A NG schematic detailing the electrical interface of the Arduino microcontroller, camera shutter and mode buttons, real-time clock, transistors, resistors, and microcontroller power source.
- *CoralCamPCB_v1.2-Simple*: Compressed (.zip) Gerber file of the CoralCam PCB in its simplest, most user-friendly format. End users can send this file directly to any PCB manufacturer of their choice to have CoralCam PCBs manufactured at low cost. **CoralCamPCB_v1.2** is the PCB version shown in this build guide, no build steps differ from construction of **CoralCamPCB_v1.2-Simple**.
- *Coral recruitment monitoring platform (CRMP) Assembly*: Photograph depicting the coral recruitment monitoring platform used as a test of CoralCam's capacity to monitor coral fragments at daily scales for extended periods.
- *CRMP_Camera_Mount*: A 3D-printable mount designed to hold CoralCam (in underwater housing) perpendicular to and above a coral settlement tile.
- *CRMP_Settlement_Tile*: A 3D-printable coral settlement tile used in the demonstration of CoralCam which fits into the CRMP base plate.
- *CRMP_Base_Plate*: A 3D-printable base plate to which the CRMP camera mount and settlement tile are fixed. This base plate is designed to be mounted to a dive weight, or directly to the substrate via a variety of through-holes.



Fig. 3. CoralCam unit prior to test deployment on a CRMP platform. Before deployment a coral fragment is fixed to the center of the settlement tile. CoralCam's underwater housing is wrapped in electrical tape to limit biofouling damage – the lens is left exposed. CoralCam and CRMP were anchored to the seafloor with a dive weight in this test.

- *CoralCam_TwiceDailyPhoto_ArduinoCode*: Code for use in the open-source Arduino Software IDE, allowing CoralCam to be programmed for twice-daily photo capture at scheduled times while entering a low-power sleep state between capture events.
- *CoralCam_TwiceDailyVideo_ArduinoCode*: Code for use in the open-source Arduino Software IDE, allowing CoralCam to be programmed for twice-daily capture of 30-second video segments at scheduled times while entering a low-power sleep state between capture events.

4. Bill of materials

Designator	Component	Number	Cost per unit - currency	Total cost- currency	Source of materials	Material type
Camera	SoCoo C30 1080P 12MP Waterproof Sports Camera with 1350 mAh battery (or similar)	1	\$19.99- \$40.00 USD	\$25.99 USD	Amazon	Other
Camera housing	Suptig Replacement Waterproof Case Protective Housing for GoPro Hero4	1	\$16.99 USD	\$16.99 USD	Amazon	Other
Camera housing "extended backdoor"	Suptig Replacement BacPac Backdoor for GoPro Hero4	1	\$8.99 USD	\$8.99 USD	Amazon	Other

(continued on next page)

(continued)

Designator	Component	Number	Cost per unit - currency	Total cost- currency	Source of materials	Material type
Camera lens protector	Tempered Glass Screen Protector for GoPro Hero4 Session, 0.3 mm 9H Hardness	1	\$2.33 USD	\$2.33 USD	Amazon	Other
Memory card	SanDisk Ultra 32 GB microSDHC UHS-I card	1	\$7.39 USD	\$7.39 USD	Amazon	Other
Arduino microcontroller	Arduino Pro Mini 3.3 V 8 MHz	1	\$2.50 USD	\$2.50 USD	EBay, Amazon, Adafruit	Other
FTDI programmer for Arduino	FT232RL FTDI Mini USB to TTL Serial Converter Adapter Module 3.3 V 5.5 V	1	\$5.99 USD	\$5.99 USD	Ebay, Amazon , Adafruit	Other
USB to miniUSB cable	AmazonBasics USB 2.0 Cable – A-Male to Mini-B – 3 Feet (0.9 m)	1	\$4.80 USD	\$4.80 USD	Amazon	Other
Transistor	BC547 NPN transistor 45 V 0.1A	2	\$0.06 USD	\$0.12 USD	EBay, Amazon, Adafruit, DigiKey	Other
Resistor	4.7 k ohm ceramic resistor 1/4 W	2	\$0.06 USD	\$0.12 USD	EBay, Amazon, Adafruit, DigiKey	Ceramic
Lithium polymer battery	3.7 V 500 mAh 802,540 single-cell lithium polymer battery	1	\$1.60 USD	\$1.60 USD	EBay, Amazon, Adafruit	Other
Ribbon cable (approximately 1 m)	5 pin/conductor ribbon cable, multiple colors	1	\$0.60	\$0.60	EBay, Amazon, Adafruit	Other
Optional CoralCam PCB	Custom printed circuit board for CoralCam	0–1	\$1.00 USD	\$1.00 USD	OSHPark, JLCPCB, or other PCB printing facility	Other
Micro JST pH 1.25 2 pin connectors (one male with wire, one female without wire)	Micro JST pH 1.25 2 PIN male female plug connector with wire cables 100 mm	1	\$0.20 USD	\$0.20 USD	EBay, Amazon, Adafruit	Other

Not included in the list Materials list is a soldering iron and solder, as these may vary depending on user preference and generally will not impact hardware function. The authors suggest a soldering iron capable of sustaining temperatures of 350–450 °C.

5. Build instructions

5.1. General safety concerns

Construction of CoralCam requires soldering of connections and use of a soldering iron. Surfaces and the iron will be hot during soldering and should be handled carefully. Stripping wires of insulation should be accomplished with a purpose-made wire stripper to avoid cutting hazards. While the voltage being used in this design does not exceed 4.2 V direct current (DC), caution should be taken to properly insulate wire leads after soldering and avoid short circuits, especially if modifying connections to the lithium polymer battery pack.

5.2. Modifying the camera

Explanation: The camera used in CoralCam is an inexpensive “action camera” that has been non-permanently modified by interfacing a microcontroller directly with the mode/power and shutter buttons. To accomplish this interface, wires are soldered to external contacts on both the Shutter button and Mode/Power button of the camera, and these wires are attached to the microcontroller. While plastic faceplates are removed from camera buttons to allow easy access for soldering, all buttons retain full functionality. Camera modifications may be undone through unsoldering the added wires and reassembly of the camera button faceplates.



Fig. 4. (a) The camera to be modified with face plate removed, (b) during removal of frame screws, and (c) after partial separation of the camera frame from camera assembly.



Fig. 5. Example of an approximately 10 cm paired wire consisting of two individual wires. One paired wire will be connected to the camera Mode/Power button and one to the camera Shutter button.

Process:

1. First remove the camera battery. Gently remove the camera's plastic face plate, which is held in place by several plastic retaining clips. Once the face plate is removed, unscrew and retain each of four small screws which hold the primary camera assembly into the plastic camera frame. Gently pry the main camera assembly away from the plastic frame until the upper shutter button contacts are fully exposed, taking care not to disconnect the ribbon cable which runs from the back of the assembly to the camera screen. Do not fully remove the primary camera assembly. Remove the plastic button cover for the shutter button from the top of the plastic camera housing, and the two plastic buttons from the side of the camera (Fig. 4).
2. Identify the shutter button on the main camera assembly top, and the mode/power button on the main camera assembly front. On most "action camera" models, these buttons consist of three metal contacts covered by a metal detent plate, usually covered with a small length of non-conductive tape. Do not remove the metal detent plate or protective tape. This style of tactile button functions such that the pairs of contacts which are farthest apart are internally connected, and a push of the button will connect the contacts which are the shortest distance apart. CoralCam functions by electronically connecting these contacts to mimic a push of the camera button.
3. Prepare two approximately 10 cm lengths of ribbon cable, each consisting of 2 wires. Strip approximately 3 mm from the end of each wire. Twist and lightly tin the ends of each wire (Fig. 5).

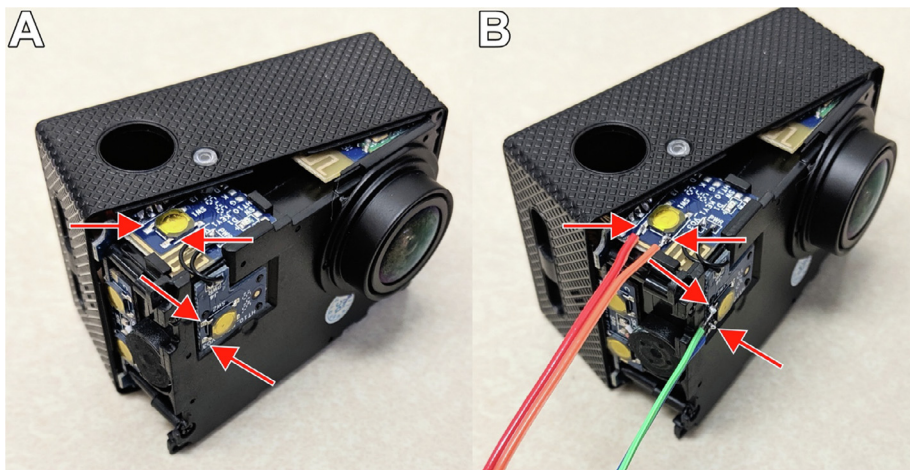


Fig. 6. (a) The camera button contacts identified before soldering and (b) after soldering.

4. Place the camera upright with the shutter button facing upward and the lens facing toward you. Heat a soldering iron to approximately 375 °C. Taking care not to melt any plastic camera components, gently place a small amount of solder on the two left-hand Shutter button contacts, and the two left-hand Mode/Power button contacts (shown in Fig. 6 with red arrows). Align the prepared tips of a paired wire with the prepared contacts of the Shutter button, placing one wire tip on the farthest left-hand contact (shown in Fig. 6 as the red wire) and one wire tip on the closest left-hand contact (shown in Fig. 6 as the orange wire). Gently press down on each wire tip with the soldering iron until the prepared button contact and wire tip solder reflows into a single mass, then remove the soldering iron while holding the wire in place until the connection cools to a solid state. Repeat this process for the two left-hand contacts of the Mode/Power button on the camera front.
5. After soldering wires to both the Shutter and Mode/Power buttons, carefully route all button wires through the uppermost hole on the side of the camera frame (Fig. 7). Pull the slack out of each wire pair, taking care not to break the soldered connection to the camera buttons. Gently push the camera assembly back into the camera frame until it clicks into position. Secure the camera assembly into the camera frame with the four screws removed in Step 1 (Fig. 8).

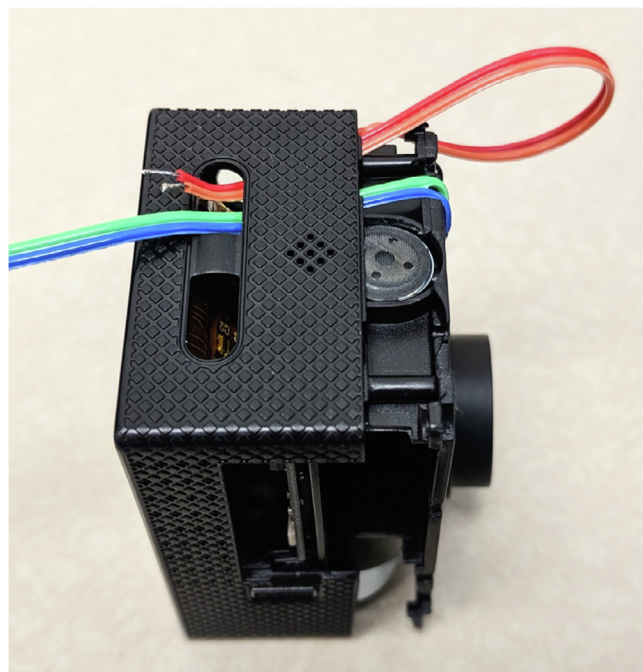


Fig. 7. Routing of camera button wires under the camera frame and out the uppermost button hole on the camera side.

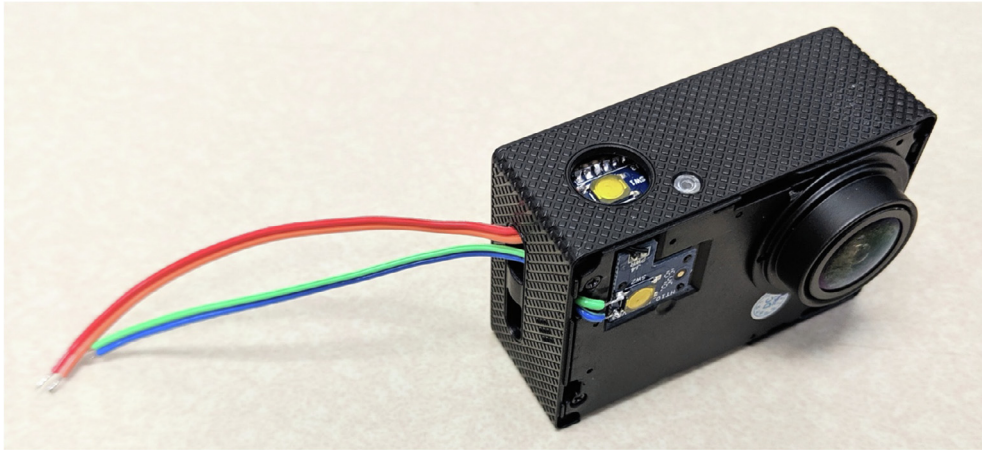


Fig. 8. The camera after completion of all modifications and proper routing of button wires.

5.3. Preparing the PCB assembly

Explanation: A power-efficient, low-cost Arduino Pro Mini 3.3 V 8 MHz microcontroller controls the camera through the use of two transistors (BC547 NPN). Scheduled image capture is accomplished through use of a real-time clock. These components require soldering to the printed circuit board (or wired via the provided schematic) to form the assembly will be interfaced with the camera through the wires soldered to the camera buttons.

Note: The printed circuit board design included in this manuscript's Source File Repository may experience minor revisions over time to improve CoralCam performance, capacity for additional instrumentation, or ease of use. These revisions will not affect the viability of this assembly guide. Release notes for PCB design revisions and an instructional build video are available in the Source File Repository.

Process:

6. Gather all components required for PCB assembly (Fig. 9). These include one Arduino Pro Mini 3.3 V 8 MHz with included header pins, one DS3231 real time clock breakout board, two BC547 transistors, two 4.7 k ohm ceramic resistors, one Micro JST pH 1.25 2 pin female connector, one Micro JST pH 1.25 2 pin connector with wires attached, and the CoralCam PCB.

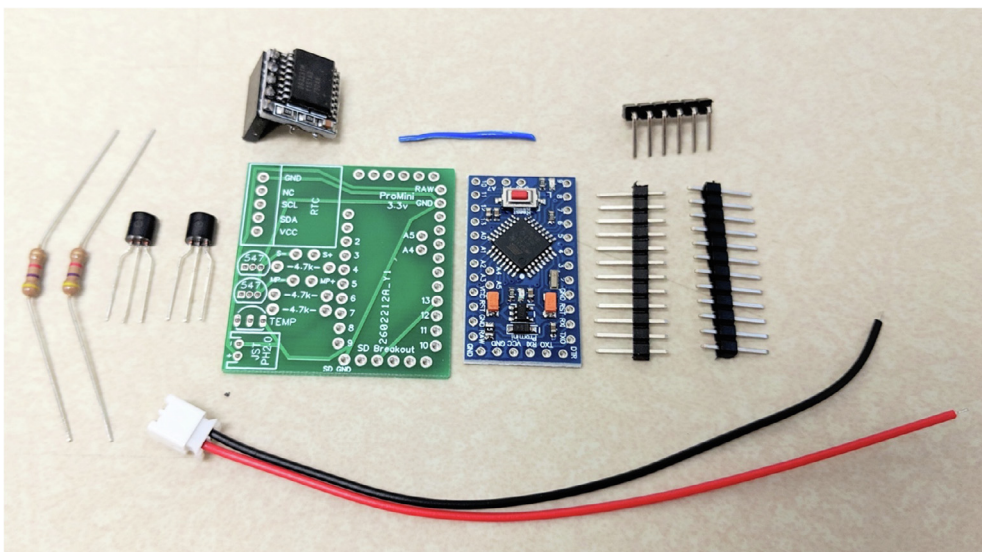


Fig. 9. Components of the PCB assembly including (2) resistors, (2) transistors, (1) real-time clock, (1) Arduino Pro Mini 3.3 V with assorted header pins, (1) female micro JST connector, (1) male micro JST connector with wires, (1) approximately 1 cm length of wire.

7. Place one long row of header pins beneath the right side of the PCB with the long edge routing upward through the PCB, and the PCB resting on the black stoppers. Use diagonal cutters to remove a set of two header pins from the unused row of header pins. Similar to the first set of header pins, place this set of two header pins through PCB holes labeled 5 and 6 on the PCB. Repeat this process to place header pins through PCB holes labeled A4 and A5. Align the Arduino with the labeled through-hole markings on the PCB and place it onto the header pins as shown (Fig. 10).

Optional: Use a small breadboard to hold header pins and the PCB assembly in place while soldering (Fig. 10).

8. Heat a soldering iron to approximately 375 °C and solder the header pins to the top surface of the labeled through-holes on the Arduino: 5, 6, RAW, GND, VCC, A4, A5. After soldering, turn the PCB over and gently remove the plastic header pin stoppers with pliers or tweezers – unused header pins will pull free with the plastic stoppers. Solder the remaining pins (5, 6, RAW, GND, VCC, A4, A5) to the bottom of the PCB. Use diagonal cutters to trim soldered header pins flush with the bottom surface of the PCB. The Arduino has now been successfully integrated into the CoralCam PCB.
9. Following the same process used to integrate the Arduino with the PCB, solder five header pins to the row of through-holes at the top of the Arduino labeled DTR, TX, RX, VCC, GND. The provided 90-degree header pins may be used if desired by the user. Programming pins only require soldering to the Arduino, and not the PCB bottom surface. Use diagonal cutters to trim header pins flush with the PCB bottom. These five pins will allow programming of the Arduino via the FTDI programmer and USB to miniUSB cable. NOTE: Soldering of programming pins is recommended but optional, these connections may be manually held in place by the user during programming.
10. Place one BC547 transistor into each location on the PCB labeled “547” by aligning the transistor face to the diagram on the PCB and gently pushing the transistor pins through the matching holes until the transistor snaps into place flush against the PCB upper surface. Flip the PCB over, solder each transistor pin to its through-hole, and use diagonal cutters to trim transistor pins flush with the PCB bottom surface.
11. Gather one 4.7 k ohm resistor and bend the leads at a sharp 90-degree angle. Locate the PCB holes labeled “4.7 k” immediately adjacent to the uppermost transistor. Route one lead of the resistor through each hole. Using pliers, pull the leads from underneath the PCB until the resistor sits flush with the upper PCB surface. Solder the resistor leads in place from the PCB bottom surface, and trim flush with diagonal cutters. Repeat this process to place and solder a resistor next to the second transistor as shown below (Fig. 11).
12. Gather the female Micro JST connector and male connector with attached wires. Using pliers, gently bend the female connector pins to an approximately 90-degree angle (Fig. 12). Insert the male connector into the female connector. Align the connector pins with the PCB area labeled “JST”, ensuring that the positive (red) wire of the male connector routes toward the PCB “+” marking for the JST connector. Solder the JST connector in place from the bottom PCB surface and use diagonal cutters to trim each pin flush with the bottom of the PCB.
13. Gather the real time clock. If your RTC has female connections pre-soldered these must be desoldered and removed. Do this by first removing the black female connector with pliers, then desoldering each of the five female pins. Remove excess solder from the RTC through-holes with a solder removal suction tool or solder removal tape. Following the same process used to attach the Arduino to the PCB, use five header pins to attach the RTC to the labeled portion of the PCB. The RTC battery should sit flush against the upper PCB surface with the RTC chip facing upwards. Ensure the RTC through-hole labels match the corresponding labels on the PCB (GND to –, VCC to +, etc.) and solder in place (Fig. 13). Similar to the Arduino, after soldering the RTC in place flip the PCB over to remove the header pin stoppers and solder the header pins to the bottom surface of the PCB. Use diagonal cutters to trim these five soldered header pins flush with the PCB bottom and top of the RTC if necessary.

Note: The RTC through-hole labeled “NC” does not require a connection. It may be soldered, or this header pin may be left unsoldered and removed as shown (Fig. 13).

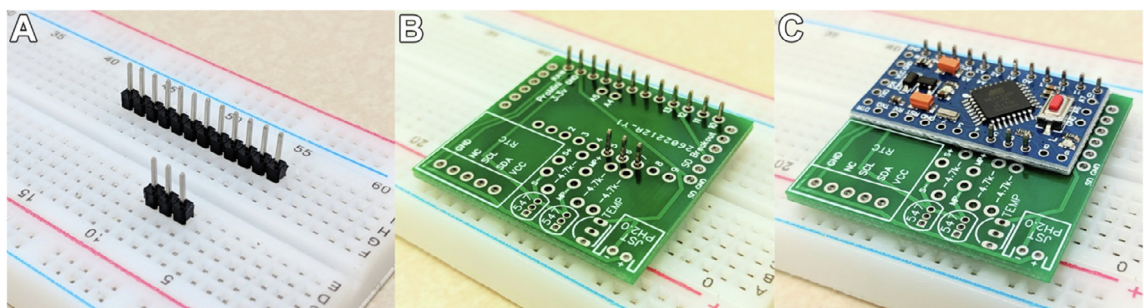


Fig. 10. (a) Demonstration using a breadboard to hold header pins in place, followed by (b) placing the CoralCam PCB over the header pins, and (c) placement of the Arduino Pro Mini for soldering.

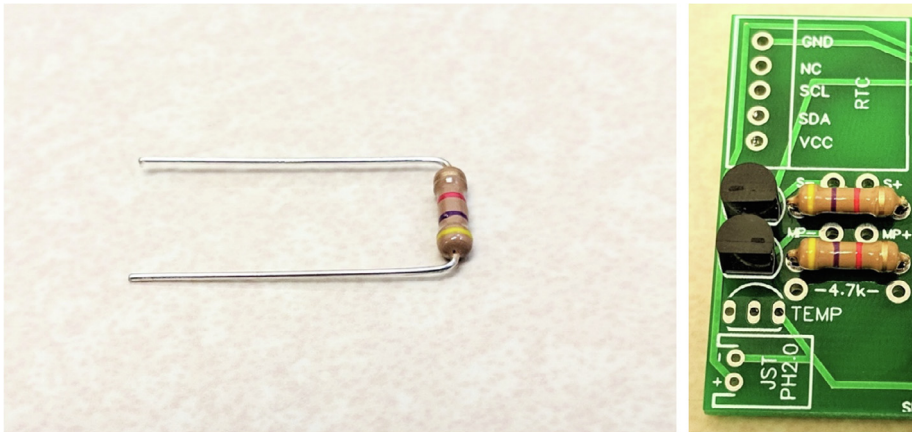


Fig. 11. Folding resistor leads and placement of the resistors and transistors on the CoralCam PCB.

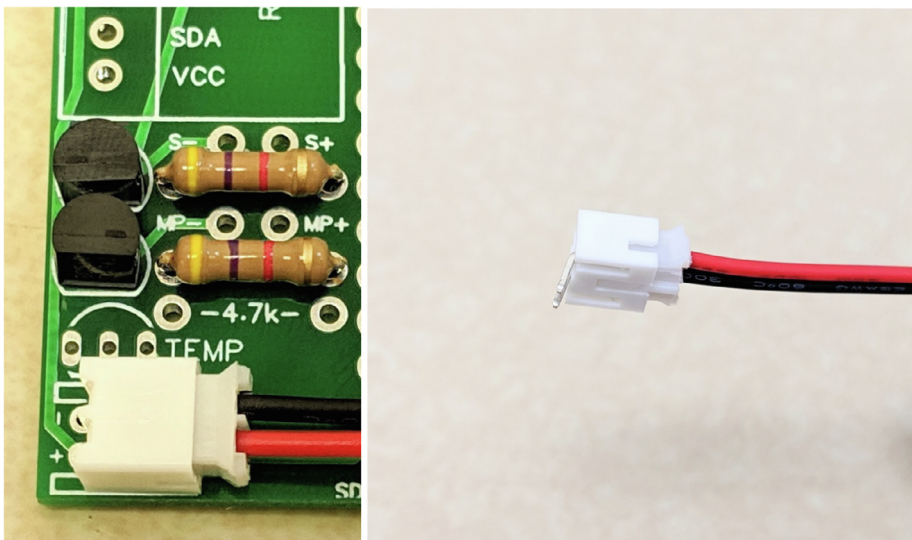


Fig. 12. Bending of the JST connector pins and positioning of the connector onto the CoralCam PCB for soldering. The positive connector wire (red) should align with the positive (+) marking on the PCB. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

14. Waking the Arduino microcontroller from low-power sleep states is accomplished via an interrupt signal from the real-time clock sent to digital pin 2 on the Arduino. Cut a 1 cm length of wire, stripping and tinning approximately 3 mm from each end. Next, rotate the PCB assembly until the small index circle (Fig. 14) on the RTC chip is facing downward and to the right. Heat a soldering iron to approximately 375 °C and gently place a small ball of solder on the third pin from the bottom, on the right-hand side of the RTC – this is the interrupt pin. Place one end of the stripped wire over the solder ball and gently press down with the soldering iron until the wire and RTC pin are connected. Take care not to overheat the RTC or bridge multiple RTC pins with solder. Place the free end of the stripped wire through the Arduino hole labeled “2” and solder in place – this connection does not need to extend into the PCB (Fig. 14).
15. The Arduino microcontroller features a green LED which is powered constantly during operation and greatly reduces battery life. This LED must be removed to accomplish CoralCam deployments greater than 20 days. The power indication LED is most easily removed using a small pair of clippers, wire cutters, or flush cutting pliers. Alternatively, remove the LED by heating a soldering iron to approximately 375 °C and placing a small ball of solder onto the tip of the soldering iron. Gently press the tip of the soldering iron against one end of the LED to desolder the contact. Repeat this process on the other end of the LED to remove the LED completely from the microcontroller (Fig. 15). The CoralCam PCB assembly is now complete, modified to be power efficient, and ready to be integrated with the camera.

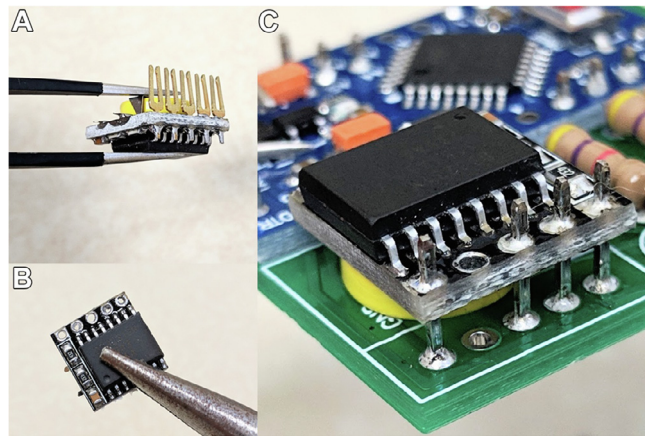


Fig. 13. (a/b) Removal of female header pins from the real time clock, (c) placement and soldering of the real time clock onto the CoralCam PCB.

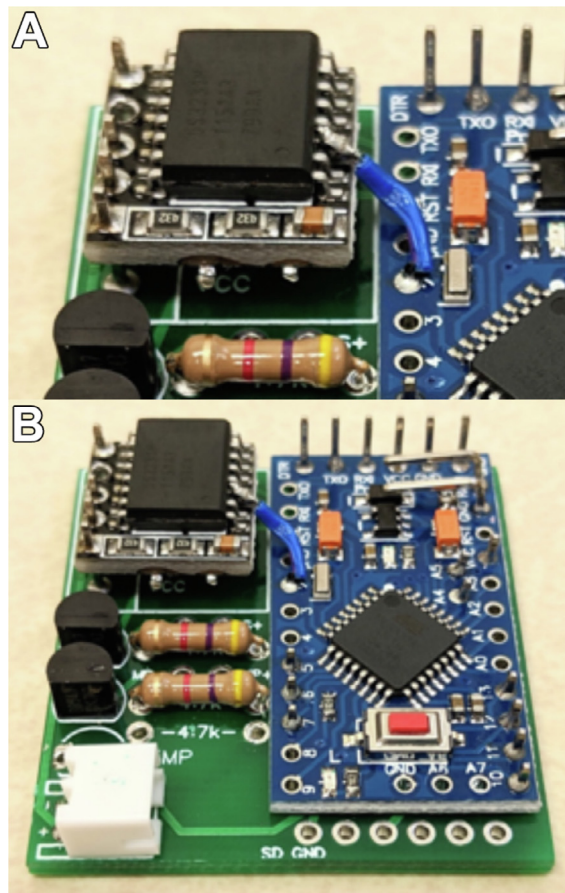


Fig. 14. (a) Positioning of a single wire soldered between the interrupt pin of the real time clock and digital pin 2 of the Arduino Pro Mini, (b) the complete CoralCam PCB assembled and ready for integration with the camera.

5.4. Wiring the camera to the PCB assembly

Explanation: Control of the action camera by the microcontroller is accomplished via the PCB transistors which electronically ground the Mode/Power and Shutter buttons. This section details the integration of the modified camera with the fully assembled CoralCam PCB.

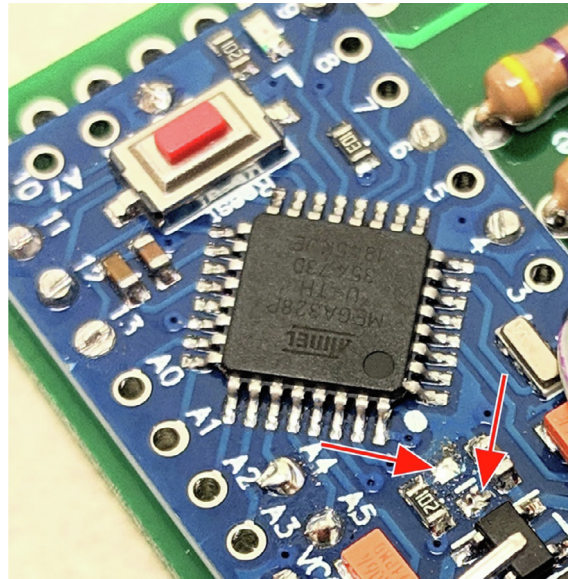


Fig. 15. The Arduino Pro Mini after removal of the power LED indicator.

Process:

16. Gather the camera which has been modified with two wires attached to the Mode/Power button, and two wires attached to the Shutter button. Approximately 3 mm at the free end of each wire should be stripped and tinned (per Step 1 of this guide). Remove the camera battery. Orient the camera upright, with the camera lens facing away from the user and the rear LCD screen facing toward the user. In this orientation the shutter button positive (+) contact is closest to the user and the shutter button negative (-) contact is farthest from the user. Similarly, the Mode/Power button positive (+) contact is uppermost on the camera front with the Mode/Power button negative (-) contact below. Route the positive (+) wire from the shutter button to the PCB through-hole labeled “Shutter +” or “S+” and solder in place (Fig. 16). Route the negative (-) wire from the shutter button to the PCB through-hole labeled “Shutter -” or “S-” and solder in place. Repeat this process for the Mode/Power button, soldering the wired connections to the appropriately labeled PCB through-holes for “Mode/Power +” or “MP +” and “Mode/Power -” or “MP -”. The camera is now integrated with the CoralCam PCB assembly (Fig. 17).

Note: Variation in manufacturing can lead to low-cost action cameras having reversed polarity vs the example provided here. After soldering the wired connections as described in Step 15 insert a fully charged camera battery. If the camera immediately powers on without pushing the Mode/Power button, then the polarity is reversed. To remedy this simply flip the Mode/Power button wiring by resoldering the Mode/Power positive (+) wire to the negative (-) Mode/Power PCB through-hole, and the Mode/Power negative wire (-) to the positive Mode/Power PCB through-hole. Re-insert the camera battery and press the Mode/Power button manually to ensure the camera powers on and off normally.

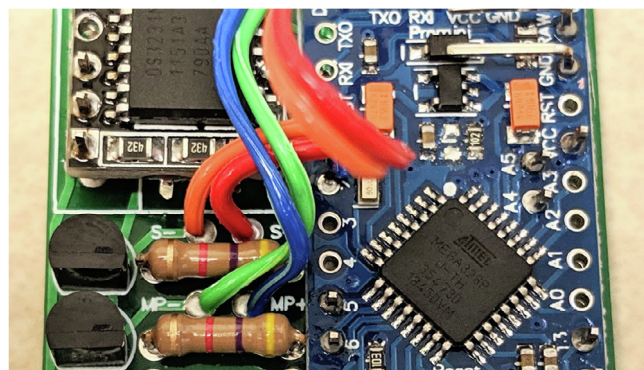


Fig. 16. Typical connection of the camera Shutter and Mode/Power button wires to the CoralCam PCB.

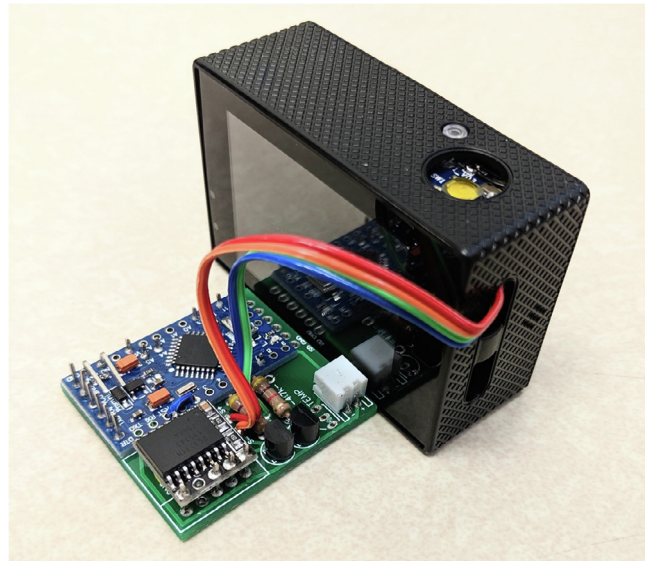


Fig. 17. A fully assembled CoralCam complete with PCB assembly and wired connections to the camera Mode/Power and Shutter buttons. Note: Tape or hot glue can be used to further limit strain on wired connections to camera.

5.5. Programming CoralCam in the Arduino IDE

Explanation: CoralCam is built around an Arduino microcontroller and is most easily programmed through the open-source Arduino Software IDE. This section describes how to load the provided code onto CoralCam prior to deployment. A full explanation of each code function is provided in the code comments, and a complete change-log of CoralCam program changes is provided in the project source file repository.

Note: Duration of CoralCam deployment until power failure is dictated by the capacity of the camera battery and the total amount of time the camera is powered on and collecting video or photo data. Deployments of 30 days or more are possible using the provided Arduino program files for twice-daily photo or 30-second video collection, but total battery duration is influenced by the temperature of deployment environment and the use state of camera batteries. Ensure all camera batteries are fully charged. If modifying the provided Arduino program files, a laboratory test of duration until power failure is recommended.

Process:

1. Download and install the free Arduino Software IDE from: <https://www.arduino.cc/en/main/software>.
2. Using five Female-Female jumper cables, connect the Arduino Pro Mini to the FTDI programmer by matching the programmer pins to the five programming pins on the microcontroller labeled “DTR”, “TX”, “RX”, “VCC”, and “GND”. Note: The FTDI programmer’s “TX” transmit pin should be connected to the “RX” receive pin on the microcontroller, and the FTDI programmer’s “RX” receive pin should be connected to the “TX” transmit pin on the microcontroller. Ensure the FTDI board jumper is set to 3.3 V, then connect the FTDI programmer to a computer with the USB-to-miniUSB cable.
3. Start the Arduino Software IDE and open either of the Arduino program files (.ino extension) provided in the Source File Repository of this manuscript. Two Arduino program files are provided: CoralCam_TwiceDailyPhoto_ArduinoCode for twice daily photo capture, and CoralCam_TwiceDailyVideo_ArduinoCode for twice daily capture of 30 s video segments. Scroll to the appropriate code section with comment heading “SET CAPTURE TIME HERE” and enter the desired twice-daily capture times. Under Sketch > Include Library select “Add .ZIP Library” and install the RTCLib code library included as a .ZIP file in the Source File Repository of this manuscript (RTCLib.h). After loading all libraries select Tools > Board and select “Arduino Pro or Pro Mini”. Under Tools > Processor select “ATMega328P 3.3 V 8 MHz”. Under “Port” select the USB port to which the FTDI programmer is connected. Select the “Compile” button to verify there are no compilation errors. If no compilation errors arise, select “Upload” to upload the CoralCam code to the microcontroller. The Arduino Software IDE will notify the user of any errors or when code has successfully uploaded to the microcontroller.
4. Disconnect the microcontroller from the Female-Female jumper cables which connect it to the FTDI programmer.

5.6. Preparing the microcontroller battery:

Explanation: The microcontroller is highly power-efficient and may be modified further for increased power efficiency. A single 500 mAh lithium polymer battery is used to power the microcontroller during field operations. These batteries may

come with no connectors or any of a wide variety soldered on. We suggest replacing these connectors with the male micro JST PH connector.

Process:

NOTE: Ensure only one battery lead wire is exposed at any time to avoid a potential short-circuit. Do not use batteries which supply above 5 V if connecting to the RAW and GND pins of the microcontroller or the PCB JST connector.

1. Strip approximately 3 mm of insulation from the tip of the red battery wire and the red wire connected to the male micro JST connector, and tin both wire tips. Slide a 1 cm segment of small diameter heat shrink tubing onto the male JST connector red wire. Solder the red battery wire tip to the red male JST connector wire tip. Slide the heat shrink tubing segment over the solder joint and seal with a heat gun to reinforce the solder joint. Repeat this process for the black battery wire and the black wire of the male JST connector. The battery is now prepared for connection to the female JST connector on the CoralCam PCB prior to deployment.

6. Operation instructions

6.1. Preparation of CoralCam for deployment

Explanation: CoralCam utilizes a low-cost “action camera” which fits easily into underwater housings designed for the GoPro Hero4. An “extended backdoor” for these housings allows space for the CoralCam PCB assembly and microcontroller battery.

Process:

1. Detach the original back door of the Hero4 housing and attach the “extended backdoor”.
2. Insert an empty microSD card and fully charged camera battery into the camera. Place the camera into the front half of the underwater camera housing.
3. Place the CoralCam PCB assembly and microcontroller battery into the “extended backdoor” of the housing. Bundle excess wiring with tape if desired.
4. Connect the microcontroller battery to the CoralCam PCB via the attached micro JST connectors. CoralCam is now powered and operating. The microcontroller should flash a red LED four times and CoralCam will execute a test capture sequence to confirm adequate power supply and that the CoralCam code is running properly.
5. Inspect the underwater housing seal o-ring for cracks and remove any debris present.
6. Gently close the underwater housing, taking care to arrange all components so that the housing closes completely, and no wires depress the red “Reset” button on the microcontroller. Ensure the housing is tightly latched shut and the watertight seal appears intact throughout the entire perimeter. CoralCam is now ready for immediate deployment.
7. *Optional:* Place a single tempered glass lens protector over the underwater housing front lens. This lens protector is not mandatory but improves the resilience of the housing lens to scratching during deployment or servicing and generally reduces the impact of biofouling.
8. *Optional:* If deploying in marine habitats, we strongly suggest wrapping all exterior surfaces of the underwater housing except for the lens in a single layer of electrical tape. This layer will act as a boundary to fouling organisms and may be removed for camera service. Do not wrap tape so tightly that exterior buttons are depressed.

6.2. Retrieval of CoralCam data and CoralCam servicing

Explanation: CoralCam may be rapidly serviced in the field for continuous deployment, or multiple CoralCam units may be rotated through stationary mounts *in-situ* to allow for increased servicing time of each unit.

Process:

1. Retrieve CoralCam from the field location and place on a dry, absorbent towel. Remove any electrical tape wrappings used and gently wipe away fouling organisms or biofilms from the underwater housing exterior using a soft sponge. Replace the front lens protector only if it appears heavily scratched after cleaning. Place CoralCam in a pool of fresh water and allow to soak for 10 min, occasionally actuating all exterior buttons.
2. Remove CoralCam from the freshwater bath and dry the underwater housing completely. Gently open the underwater housing, taking note of any apparent rust on hinges or exterior surfaces.
3. Disconnect the battery connector from the CoralCam PCB. Connect a powered microUSB cable to the camera’s microUSB port to begin charging the camera battery, or exchange with a previously-charged camera battery. Recharge the microcontroller battery with an approved lithium-polymer battery charger or replace in the field with a recharged battery. Remove the microSD card from the camera and download CoralCam data.

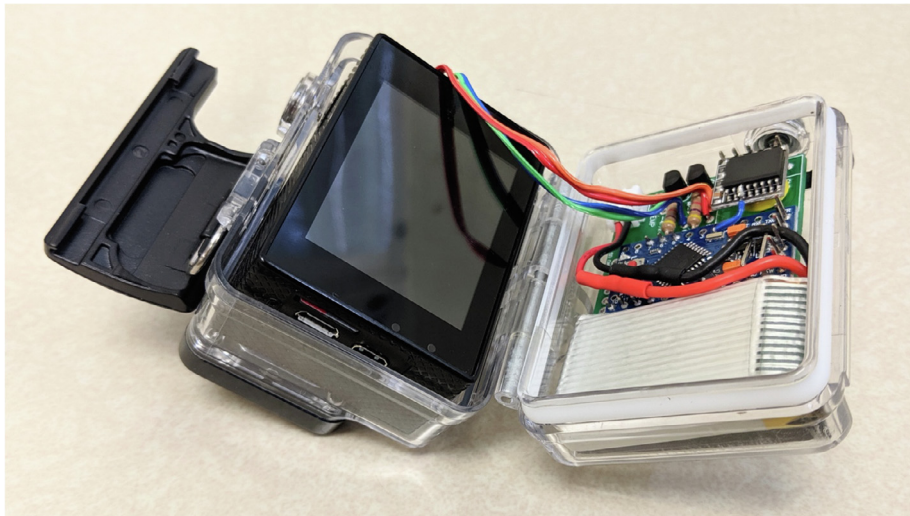


Fig. 18. A fully assembled CoralCam with microcontroller battery placed in a waterproof housing prior to sealing and deployment in the field.

4. If a new capture schedule or capture type (photo vs video) is required, upload modified code to CoralCam via the Arduino Software IDE.
5. Replace the now-empty microSD card, a charged camera battery, and a charged microcontroller battery into the CoralCam housing. Connect microcontroller battery leads to the CoralCam PCB (Fig. 18).
6. Seal the CoralCam housing and cover with electrical tape as during initial deployment. The microcontroller will flash its red LED four times and execute a capture sequence to verify adequate power and properly running CoralCam code.

7. Validation and Characterization

The hardware presented (CoralCam) is demonstrated with the collection of daily imagery of a coral fragment for a 2-week period, followed by a rapid same-day service and immediate redeployment for 30 days to continue observations. This cycle demonstrates the repeatability of CoralCam across rapid service periods, and the ability of CoralCam to capture imagery limited only by local biofouling accumulation rates for durations of one month or less. As a worse-case scenario CoralCam was placed in a shallow, heavily sedimented and nutrient rich location in Kāneʻohe Bay, Oʻahu. In this use-case a mostly 3D-printed assembly was constructed to hold CoralCam in a downward orientation over a 3D printed coral settlement plate, both of which are fixed to a 3D-printed base plate which is anchored with a diving weight. In this arrangement CoralCam is mounted approximately 20 cm above the coral fragment, just outside of the optimal minimum focusing distance of 25–30 cm when used in water. Despite this placement, imagery is sufficiently sharp with post-processing adjustments to make fine-scale measurements of coral growth or behavior and daily variation in sediment stress. Future deployments will place the camera approximately 25 cm above the subject to ensure optimal image sharpness. Preliminary tests have also indicated that the addition of an inexpensive 10× magnification “close-up” lens can provide sufficiently sharp images at placements closer than 25 cm to the subject of interest. No ingress of water into the underwater housing or rust was detected in any



Fig. 19. (left) A CoralCam with biofouling immediately following 30 days of deployment, (right) and after removal of the electrical tape wrapping and gentle scrubbing of the front lens.

deployment, and all accumulated biofouling was removed without damage to CoralCam (Fig. 19). After 30 day deployments biofouling consisted primarily of thin biofilms, algal turf, and tubeworms. The majority of biofouling occurred on the electrical tape wrapping used to protect the CoralCam underwater housing and was removed with removal of this wrapping. Fouling occurring on the lens (equipped with a tempered glass lens protector) was easily removed when gently scrubbed with a damp towel. Intensity and type of biofouling encountered will vary with deployment duration and location.

In both the 2-week test and subsequent 30-day test CoralCam consistently captured imagery of the coral fragment at two scheduled times per day. No scheduled capture periods were absent in any field test. The microcontroller successfully powered off the camera between capture events, allowing sufficient camera battery life to continue image capture for over two weeks after conclusion of the coral observation field test, suggesting longer deployments may be possible. Biofouling did not significantly occur during the initial 2-week test, and only began to significantly impact imagery after approximately 25 days of deployment under these worse-case conditions. Reduced biofouling rates are expected on deeper or more oligotrophic reefs. In these tests CoralCam demonstrated a capacity to capture a wide variety of biological and environmental processes occurring at daily or sub-daily scales. Sedimentation stress on the coral fragment, a critical driver of coral health, is visually observed in photographs to be highly variable from day to day (Fig. 20). The coral fragment can be seen reacting to these sediment stress pulses by rapidly shedding sediment-laden surface mucus layers. Multiple interactions with benthic organisms were also captured including association of a small blenny with the coral fragment and removal of sediment from the immediate area by the sea cucumber *Ophiodon elongatus* (Fig. 21). In addition, the tissue on the coral fragment can be observed swelling and contracting over the course of the deployment. Subsequent video-capture deployments outside of a shallow agro-ecological fishpond (He'eia fishpond, Kāne'ohe Bay, O'ahu) are included via stills taken from 30-second video observations, demonstrating CoralCam's detection of both human-avoidant species (Green Sea Turtle) and fish behavior (Fig. 22) – video files for each of these examples are included in the Source File Repository. In all video tests CoralCam successfully captured the scheduled 30-second video observation at the prescribed time and experienced the same biofouling results as prior photo capture deployments. The fine-scale biological interactions and indicators of coral health observed in these video and photo tests are all examples of valuable data which would likely have been missed without observations from CoralCam.

Capabilities of CoralCam

- Relatively simple construction, programming, and deployment
- Low-cost (<\$80.00 USD) compared to commercial solutions (\$589.00–\$715.00 USD)
- Requires no surface-mount or advanced soldering techniques to wire by hand. Soldering is further simplified with use of freely available CoralCam PCB schematic.
- Reliable scheduled, high-resolution video or photo capture at daily temporal scales for a minimum of 1-month deployment time.

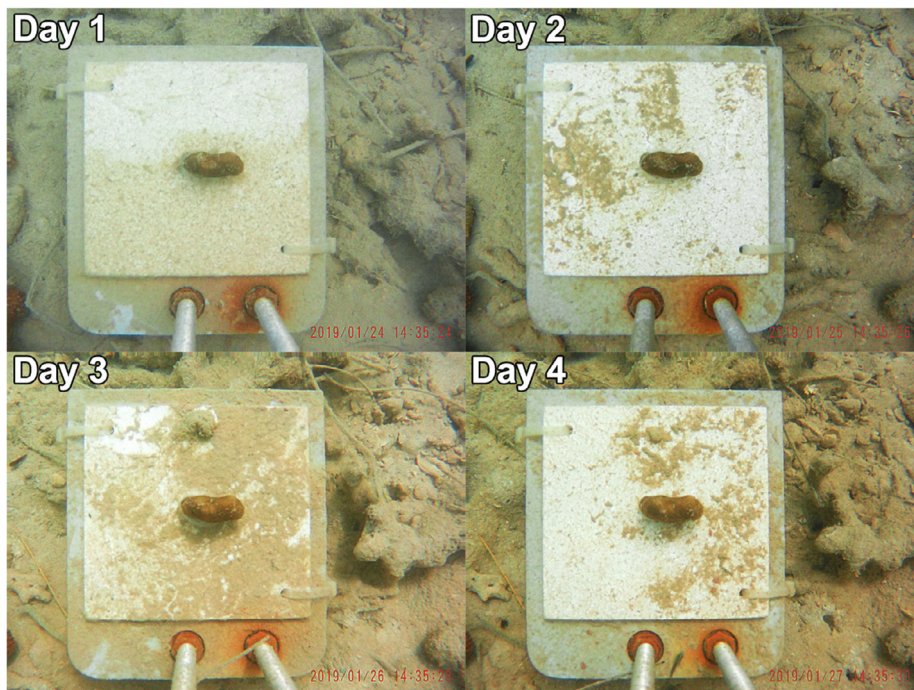


Fig. 20. Images from four consecutive days during CoralCam test deployment. Images demonstrate the utility of daily-scale observations to capture high daily variation in processes such as sedimentation stress which impact coral health.

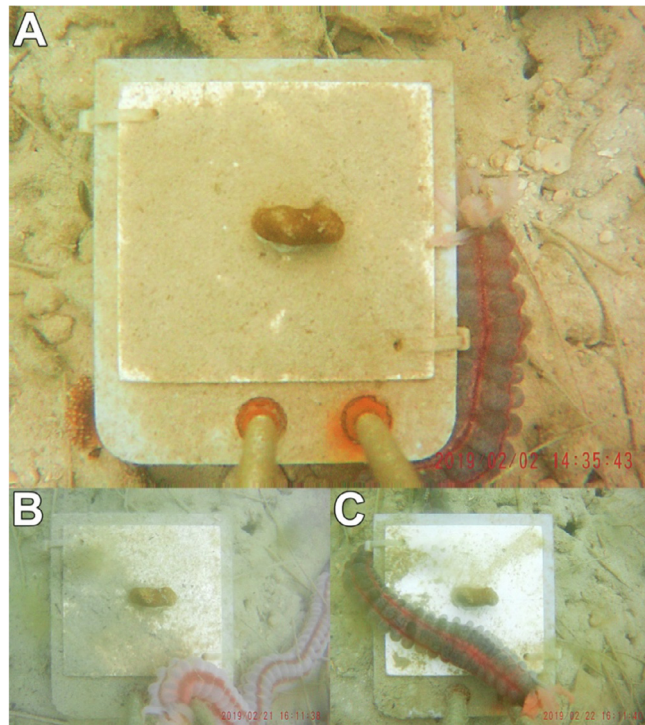


Fig. 21. Images captured during CoralCam test deployment showing daily interactions of the sea cucumber *Opheodesoma spectabilis* with a coral fragment. Images captured in 2019 on February 2 (a), February 21 (b), and February 22 (c).



Fig. 22. (top) Still taken from a 30-second CoralCam video demonstrating video observation of a Green Sea Turtle outside of a shallow agro-ecological fishpond, (bottom) still taken from a 30-second CoralCam video demonstrating observation of fish fishing behavior around a manmade structure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Demonstrated ability to resolve fine-scale biological interactions and environmental stressors, such as sediment stress on small corals.
- Data collection may be synchronized across CoralCam units via integrated real-time clock
- Easily reprogrammed for any combination of capture mode (photo or video) or schedule.
- Arduino microcontroller may be easily integrated with additional environmental sensors, UV-C biofouling control, or data-logging capabilities.
- Immediately deployable to 45 m in prescribed underwater housing. Potential for longer or deeper deployments with improved housings or larger external battery packs.
- Rapid servicing and redeployment, or easy rotation of separate CoralCam units in the field for continuous daily image capture over extended time periods.
- Flexible method applicable to control other cameras types with similar architecture. Not dependent on specific GoPro models, unlike commercial solutions. Camera modifications are non-permanent.

Limitations of CoralCam:

- Not a “plug and play” solution, some soldering required
- No wireless interface with the user, unlike available commercial solutions
- Some programming required by the end user
- Deployment duration in marine conditions limited by local biofouling accumulation rates, followed by dissipation of the onboard camera battery.

Declaration of interest

The authors declare that there is no conflict of interest regarding the publication of this article.
Human and animal rights

Acknowledgments

Hawaii Institute of Marine Biology (HIMB) contribution number 1773. University of Hawaii School of Ocean and Earth Science and Technology (SOEST) contribution number 1773. Material support for this project was provided in-part through NOAA Coral Reef Conservation Program award #NA18NOS4820110.

References

- [1] A.C. Burton, E. Neilson, D. Moreira, A. Ladle, R. Steenweg, J.T. Fisher, et al, Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes, *J. Appl. Ecol.* 52 (3) (2015) 675–685, <https://doi.org/10.1111/1365-2664.12432>.
- [2] A.W. Bicknell, B.J. Godley, E.V. Sheehan, S.C. Votier, M.J. Witt, Camera technology for monitoring marine biodiversity and human impact, *Front. Ecol. Environ.* 14 (8) (2016) 424–432.
- [3] C. Doropoulos, G. Roff, M.S. Visser, P.J. Mumby, Sensitivity of coral recruitment to subtle shifts in early community succession, *Ecology* 98 (2) (2017) 304–314, <https://doi.org/10.1002/ecy.1663>.
- [4] A.A. Monroe, M. Ziegler, A. Roik, T. Röthig, R.S. Hardenstine, M.A. Emms, M.L. Berumen, In situ observations of coral bleaching in the central Saudi Arabian Red Sea during the 2015/2016 global coral bleaching event, *PLoS ONE* 13 (4) (2018) e0195814.
- [5] I.B. Kuffner, B.H. Lidz, J.H. Hudson, J.S. Anderson, A century of ocean warming on Florida Keys coral reefs: historic in situ observations, *Estuaries Coasts* 38 (3) (2015) 1085–1096.
- [6] J.H. Connell, T.P. Hughes, C.C. Wallace, J.E. Tanner, K.E. Harms, A.M. Kerr, A long-term study of competition and diversity of corals, *Ecol. Monogr.* 74 (2) (2004) 179–210, <https://doi.org/10.1890/02-4043>.
- [7] H.S. Lenihan, S.J. Holbrook, R.J. Schmitt, A.J. Brooks, Influence of corallivory, competition, and habitat structure on coral community shifts, *Ecology* 92 (10) (2011) 1959–1971, <https://doi.org/10.2307/23034829>.
- [8] P.K. Dunstan, C.R. Johnson, Spatio-temporal variation in coral recruitment at different scales on Heron Reef, southern Great Barrier Reef, *Coral Reefs* 17 (1) (1998) 71–81, <https://doi.org/10.1007/s003380050098>.
- [9] J.A. Ahumada, C.E. Silva, K. Gajapersad, C. Hallam, J. Hurtado, E. Martin, D. Sheil, Community structure and diversity of tropical forest mammals: data from a global camera trap network, *Philos. Trans. R. Soc. B Biol. Sci.* 366 (1578) (2011) 2703–2711.
- [10] E.M. Soininen, I. Jensvoll, S.T. Killengreen, R.A. Ims, Under the snow: a new camera trap opens the white box of subnivean ecology, *Remote Sens. Ecol. Conserv.* 1 (1) (2015) 29–38.
- [11] C. Carbone, S. Christie, K. Conforti, T. Coulson, N. Franklin, J.R. Ginsberg, et al., The use of photographic rates to estimate densities of tigers and other cryptic mammals, in: *Animal Conservation forum* (vol. 4, No. 1, pp. 75–79). Cambridge University Press, 2001.
- [12] M.J. Kelly, Design, evaluate, refine: Camera trap studies for elusive species, *Anim. Conserv.* 11 (3) (2008) 182–184, <https://doi.org/10.1111/j.1469-1795.2008.00179.x>.
- [13] M. Cappel, P. Speare, G. De'ath, Comparison of baited remote underwater video stations (BRUVS) and prawn (shrimp) trawls for assessments of fish biodiversity in inter-reefal areas of the Great Barrier Reef Marine Park, *J. Exp. Mar. Biol. Ecol.* 302 (2) (2004) 123–152.
- [14] C.J. Walton, N.K. Hayes, D.S. Gilliam, Impacts of a regional, multi-year, multi-species coral disease outbreak in Southeast Florida, *Front. Mar. Sci.* 5 (2018), <https://doi.org/10.3389/fmars.2018.00323>.
- [15] Arduino AG. Getting started with the Arduino Pro Mini. Retrieved from <https://www.arduino.cc/en/Guide/ArduinoProMini>. Accessed: April 20, 2019.
- [16] CamDo Solutions. Long Term Time Lapse product collection. Retrieved from <https://cam-do.com/collections/long-term-time-lapse>. Accessed: April 20, 2019.
- [17] L.P. Miller, TimeLapse Arducam. Retrieved from https://github.com/millerlp/Time_lapse_arducam. Accessed: April 29, 2018.