Applied Measurement & Control SS2023: Environmental monitoring with a smart bird house by:

Attallah, Osama Haiyl Attallah (27355), Santina Jarkass, Ismail (25180) and Ojukwu, Henrydon-Onyeka (28057)

1.0 Introduction (Henrydon)

Due to massive urbanization efforts in cities and increasing demand for housing, industry and intensive agricultural approaches, there exists a need to compete for space against other species. The rise of unsustainable anthoropoligcal activity has been deemed a threat upon animal and plant populations, in addition to the disruption of food chains (particularly the declining insect populations that birds feed on to survive) which has put reportdly compromised the biodiversity of our ecosystems. (1) According to NABU (Naturschutzbund), bird inhabitants and species diversity in Germany are at risk and steadily declining. (2) Accordig to Fig. 1, the trend for forest, common and farmland bird species is shown. The number of forest birds remained almost stable between 1990 and 1993, then a declining trend was observed until 2009. Common and farmland avian species had a steadier trend of decreasing numbers. 15 percent of all bird breeding pairs in Germany disappeared between 1998 and 2009; in 12 years 12.7 million breeding pairs were lost. (3) Even though this has not affected all bird groups and breeding pairs with the same magnitude, there still is a necessity for environmental scienticts and protection unions to intervene and monitor the biodiveristy of birds around urban areas and living species. Due to the advancement of technology in the past decades, in addition to expanded production of realtively inexpensive sensors, microprocessors etc., monitoring devices and systems have been revolutionzed and spread around vastly, which makes it easier to observe changes in the equilbirium of nature, and accelerate further research efforts that make amends through proper environmental management and aim to reverse the disturbance caused to nature's dynamics. The objective of this project is to design and test an environmental monitoring system to monitor avian species in Kamp-Lintfort.

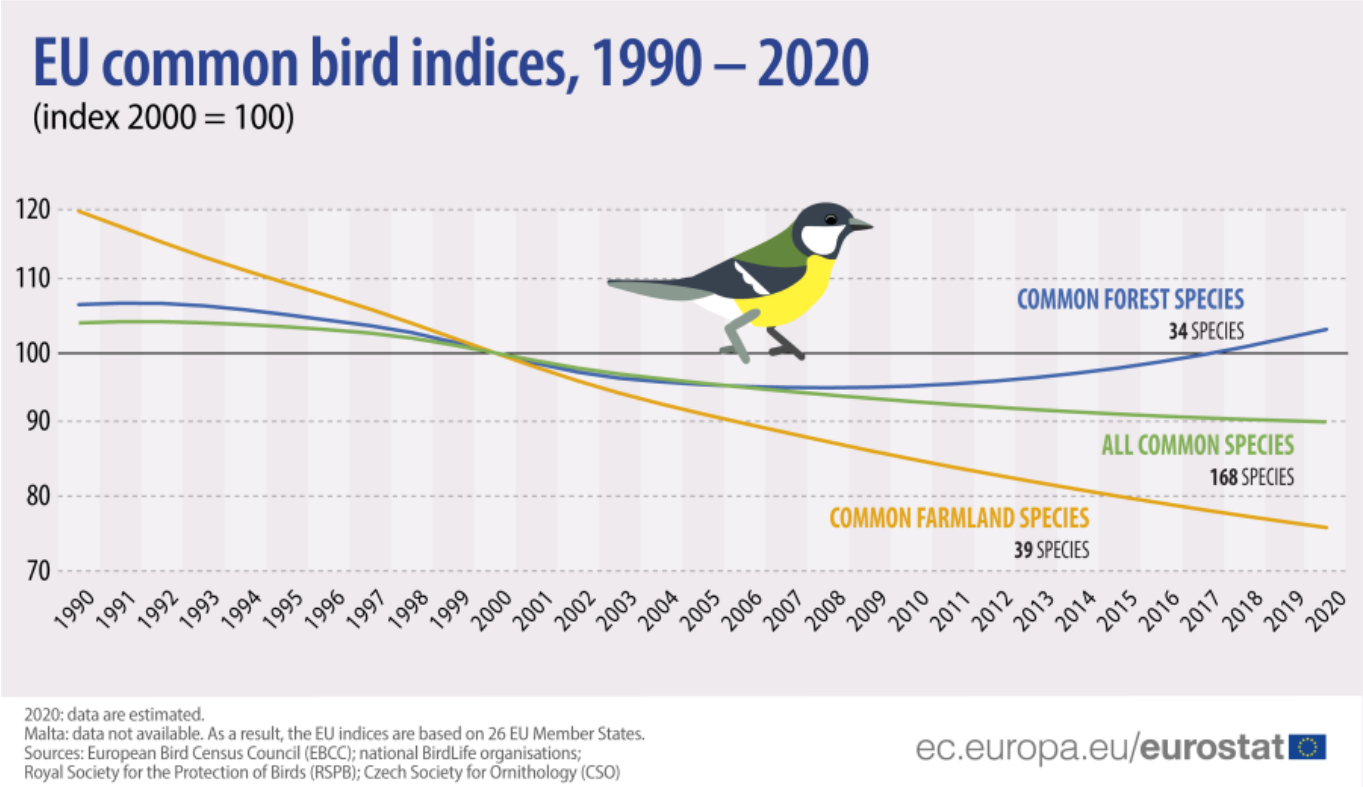


Fig 1: EU common bird indices

2.0 Materials and Methods

==Quantitaive methods of biodiversity determination== (Ismail)

To describe species diversity in natural communities, ecologists categorize several indices which based on what they measure and what they represent. No specific weights are assigned to species, except for abundances (and for biomass in some indices). The same holds for the individuals within a species. Species richness measures of biodiversity by counting the number of different species in a given area. This measure is strongly dependent on sampling size and effort, but does not depend on each species weight in the sample pool. Richness-Evenness (or Richess-Abundance) indices measure in a way similar to the information theory concerning a code or message; by accounting for the weight of each species in the sample pool. A prime example of this is the Shannon-Wiener diversity index, expressed as such:

H'= -∑pi x ln(pi)

It is calculated by first determining **pi**, which is the ratio of individual species' occurences to the total number of species' occurences, Every pi is then multiplied with its natural logarithm and then summed up. (4)

One shortcoming with the monitoring system is that it allows limited observation of the number of same-species i.e. it is difficult to notice repeated occurences of the same individual bird of a species (if, it visits the birdhouse more than once), and thus the sample size is cannot be precisley ascertained. This is explained in Table 1, For the sake of this experiment, unless clearly visible (or controlled by file metadata and data logs to the server), individual occurences of a bird species will

not be assumed to be recurring. Further, this means that the photos collected can indicate species richness, but cannot accurately give results for species evenness or their distribution in a sample area

2.1 components

Through the intelligent capabilities of the birdhouse, communication and connecvitivty with the system can be simplified, to gather data and obtain insight into remote areas without directly controlling or maintaing the system following a scheduele. Of course, there should be an incentive to ensure the sustainability of this system. For the system to be categorized as environmentally sustainable, it has to be long-term obtain energy from a renewable resource. For it to fulfill the economical sustainability criterum, it has to be inexpensive to design and implement. The framework is based on the following materials and components:

2.1.1 ESP32CAM + SD card (Ismail)

The ESP32-CAM, as shown in Fig 2 is a compact and energy-efficient camera module built around ESP32. Equipped with the OV2640 camera, it boasts an onboard TF card slot. One of its standout features is the 4MB PSRAM, which efficiently stores camera images, enabling smooth video streaming and other processes without overwhelming the ESP32, thus allowing for higher picture quality. Additionally, the module includes an onboard LED for flash functionality and multiple GPIOs for seamless peripheral connections. Users can conveniently insert an SD card to preserve the captured photos for future review. (5)

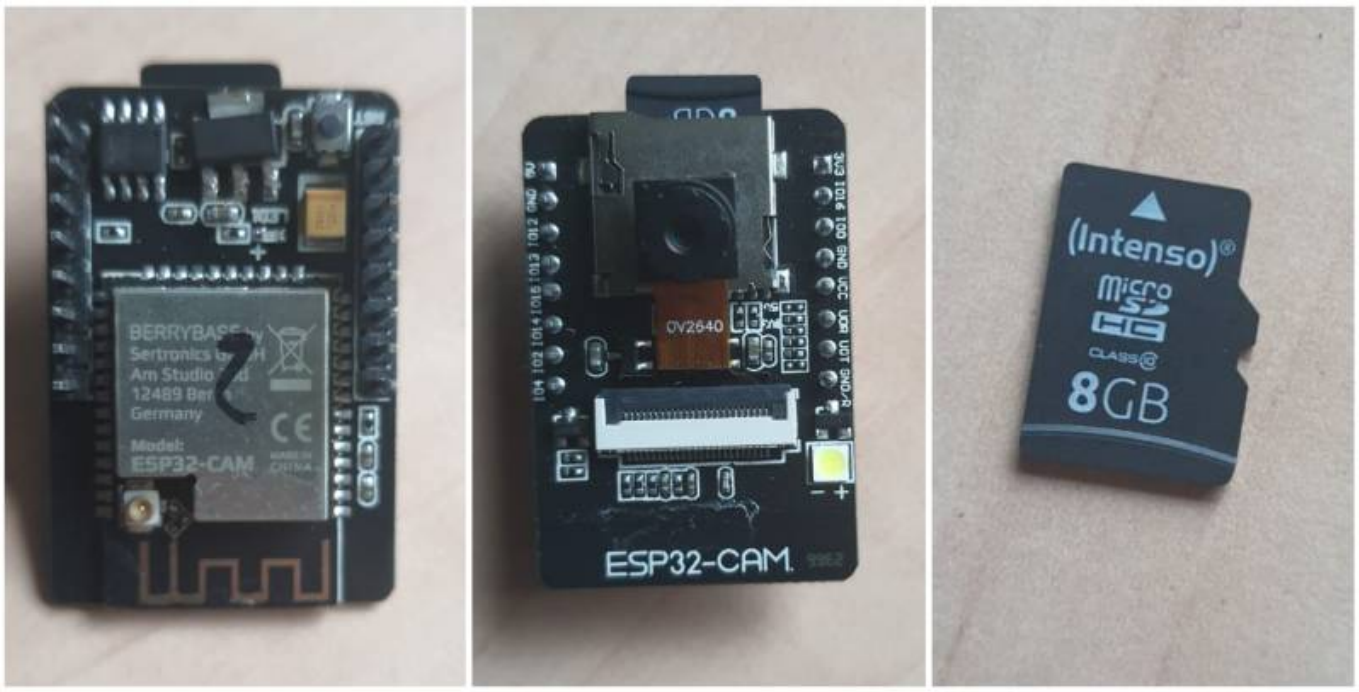


Fig 2. ESP32-Cam AI-Thinker + 8Gb SD Card

2.1.2 FT232R UartSBee V5 (FTDI) (Henrydon)

Since the ESP32-CAM AI-Thinker lacks a built-in programmer and a serial port, it cannot be

programmed directly, thus serial communication must be first established through a protocol. The FT232R is a single chip USB to serial UART interface with advanced features, including integrated USB termination resistors, EEPROM for storing device descriptors, support for various data transfer rates, FTDI's royalty-free Virtual Com Port and Direct drivers, and compatibility with different voltage levels, making it suitable for a wide range of applications. (6)

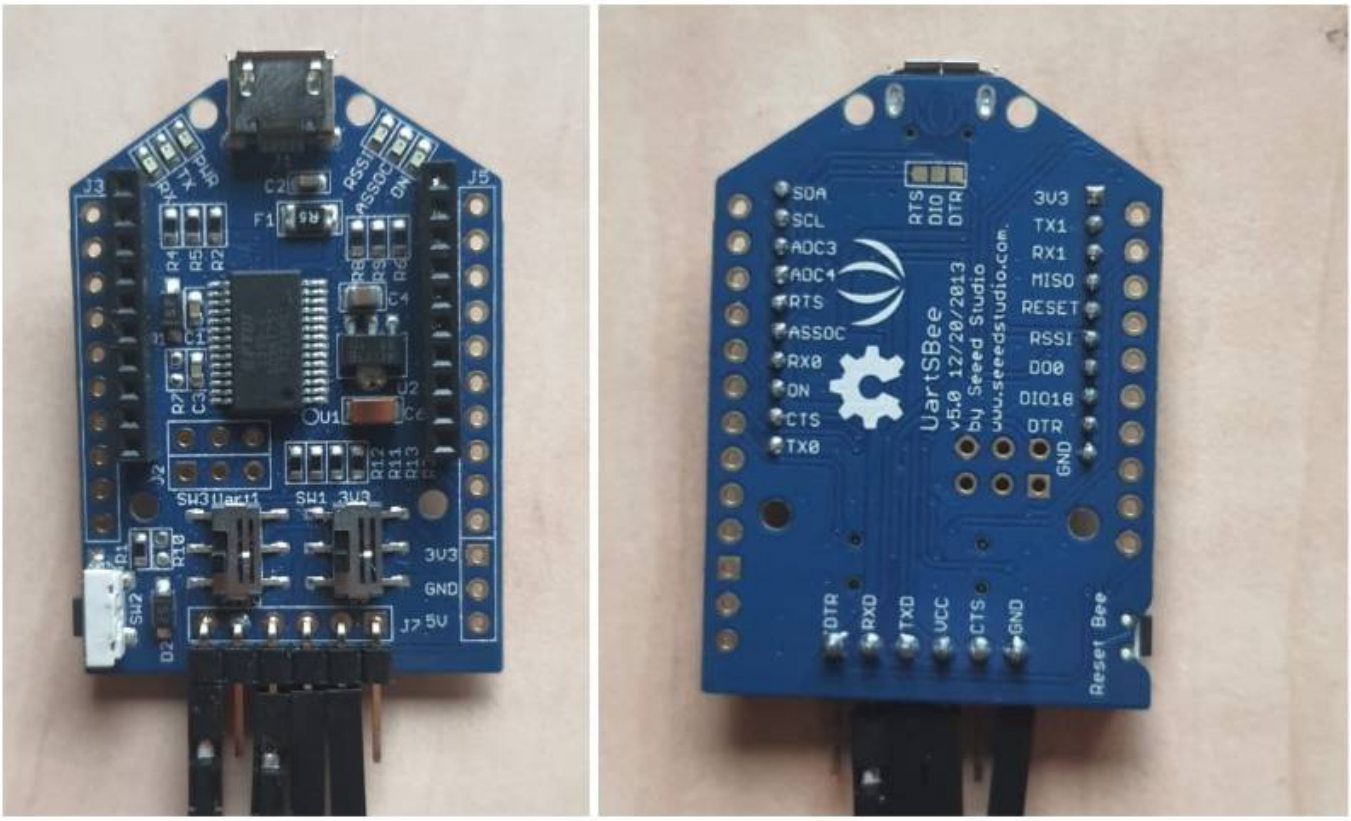


Fig 3. UartSBee V5

2.1.3 Solar module (Ismail)

Fig 4. Solar panel module

On the roof of the bird house lies a solar (or photovoltaic panel) module, shown in Fig 4. This eco-friendly component absorbs sunlight to generate clean and sustainable energy, which is then stored in a built-in rechargeable battery. This ensures a continuous and uninterrupted power supply for the bird house camera, even during cloudy days.

The short-circuit current is the current through the solar cell when the voltage across the solar cell is zero (i.e., when the solar cell is short circuited). Usually written as ISC, the short-circuit current is shown on the IV curve below. The short-circuit current is due to the generation and collection of light-generated carriers. For an ideal solar cell at most moderate resistive loss mechanisms, the short-circuit current and the light-generated current are identical. Therefore, the short-circuit current is the

largest current which may be drawn from the solar cell. (7)

Short circuit current ~1 A

The open-circuit voltage, VOC, is the maximum voltage available from a solar cell, and this occurs at zero current. The open-circuit voltage corresponds to the amount of forward bias on the solar cell due to the bias of the solar cell junction with the light-generated current. The open-circuit voltage is shown on the IV curve below. (8)

Measuring the potential difference between the positive and negative terminals of the solar module, as in Vid 1. yields a DC voltage of ~6V, which means it consists of 12 PV cells, with 0.5 V each connected serially.

panelvoltage.mp4

Vid 1. The open-circuit voltage ~ 6V

Its role in this project is to keep the battery charged at all times to ensure the constant recording done by the ESP32-CAM. Logically, the bird house is preferably placed at an elevated location, where it has direct contact to sunlight (a rooftop for instance).

2.1.4 PIR module (infrared module) (Henrydon)

PIRs are basically made of a pyroelectric sensor, which can detect levels of infrared radiation. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low. The sensor has a wide input voltage range (4. 5V to 12V), a High/Low output voltage of 3. 3V TTL, capable of distinguishing between object and human movement, featuring two operating modes, covering a 120° angle and a 7-meter range, with low power consumption (65mA) and an operating temperature range of -20° to +80° Celsius. Below the frensel lenses, the IR sensor is housed in a hermetically sealed metal can to protect against intruding noise/temperature/humidity immunity. There is a window made of IR-transmissive material that protects the sensing element. Behind the window are the two balanced sensors. (9)

Fig 5. PIr sensor

2.1.5 Resistors + Transistors

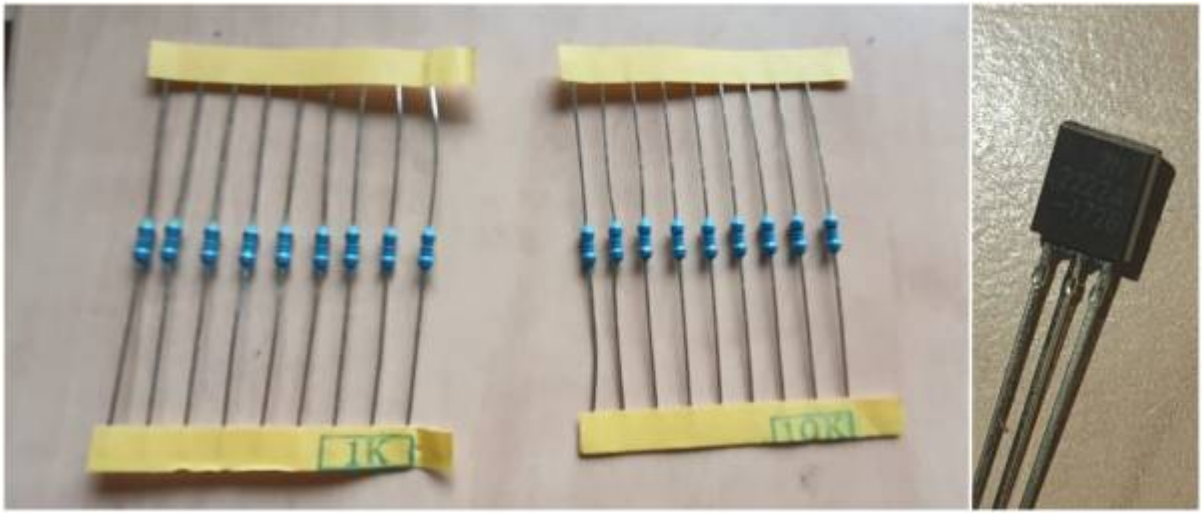


Fig 6. 1K, 10K resistors and 2N2222A NPN Transistor

The transistor here acts as a switch (10)

2.1.6 Battery + charge regulator (henry)

Lithium-ion polymer batteries are thin, light, and powerful with an output range of 3.7V to 4.2V and a capacity of 2000mAh. The battery comes with a pre-attached genuine 2-pin JST-PH connector preventing snags, smooth insertion, and removal, as well as built-in protection circuitry to prevent overcharging, overuse, and protect against output shorts. (11)

The Adafruit Universal USB/DC/Solar Lithium Ion/Polymer Charger, shown in Fig 7. is a multifunctional charging device designed to efficiently and reliably charge lithium-ion/polymer batteries using USB, DC power sources, or solar panels, catering to a wide range of portable and renewable energy applications. It includes status indicators and protection features like overcharging and reverse polarity protection, ensuring safe and efficient charging for the battery and connected devices. Functions with 3.7V/4.2V LiIon batteries, and a 6V solar panel (2.1 mm DC Jack)

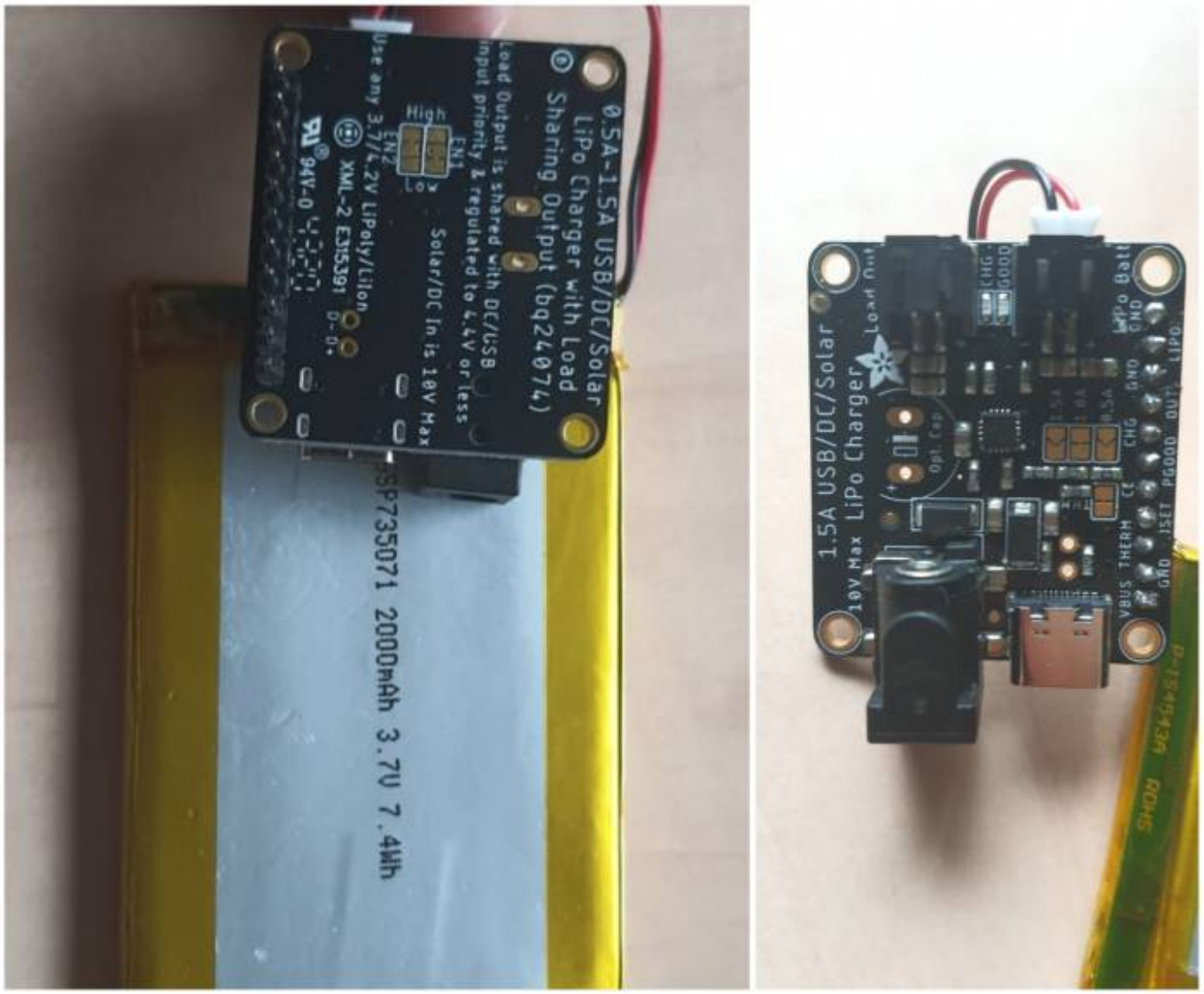


Fig 7. Battery + LiPo Charger

The bq24074 which powers this design is great for solar charging, and will automatically draw the most current possible from the panel in any light condition with a near-MPPT characteristic

Maximum Power Point Tracking is a family of control algorithms that aims at optimizing the use of a power source that possesses a fluctuating power profile. The cloud coverage significantly impacts the capability of a panel to deliver electricity. As such, maximizing the extracted power requires identifying – and tracking – the operating point that provides the highest power level as a function of the operating conditions. MPPT is applied in renewable energy systems. Other special operating points may be interesting to track, such as the maximum efficiency point tracking (MEPT), reasons related to operating costs. The inevitable internal resistance can hinder the maximum possible output power. Since the maximum power point is not located at the maximum (Voc) voltage and max. Isc current point. Therefore, the operating point that delivers the maximum power must be constantly tracked by searching for the best voltage · current combination. Voltage collapses during high current draw. We need to find a way to keep the lipo charger from drawing too much current, and backing off when the voltage starts to drop. The bq24074 is equipped with an Input Dynamic Power Managemtn Mode (Input DPM) and basically, when the input drops below 4.5V approximately, the charger will automatically increase/reduce the charge rate to keep the voltage higher than 4.5V. This charger is dynamically stable and does not need an optional capacitor to keep solar charging from oscillating.

The charger has 2 LEDS:

OUT - Regulated load output. This pin will provide a regulated output when the input voltage is below the over voltage protection threshold and above the regulation voltage. It will never be higher than 4.4V (but it may dip down to 3V or whatever the LiPo battery voltage is at, if USB/DC isnt plugged in)

PGOOD - Power Good Status (active low). PGOOD pulls to GND (open drain) lighting the connected led when a valid input source is connected. If the input power source is not within specified limits, PGOOD is disconnected from ground (high impedance) and the LED will be off. CHG - Charge status (active low) pulls to GND (open drain) lighting the connected led when the battery is charging. If the battery is charged or the charger is disabled, CHG is disconnected from ground (high impedance) and the LED will be off. (12)

2.2 schematic (osama)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

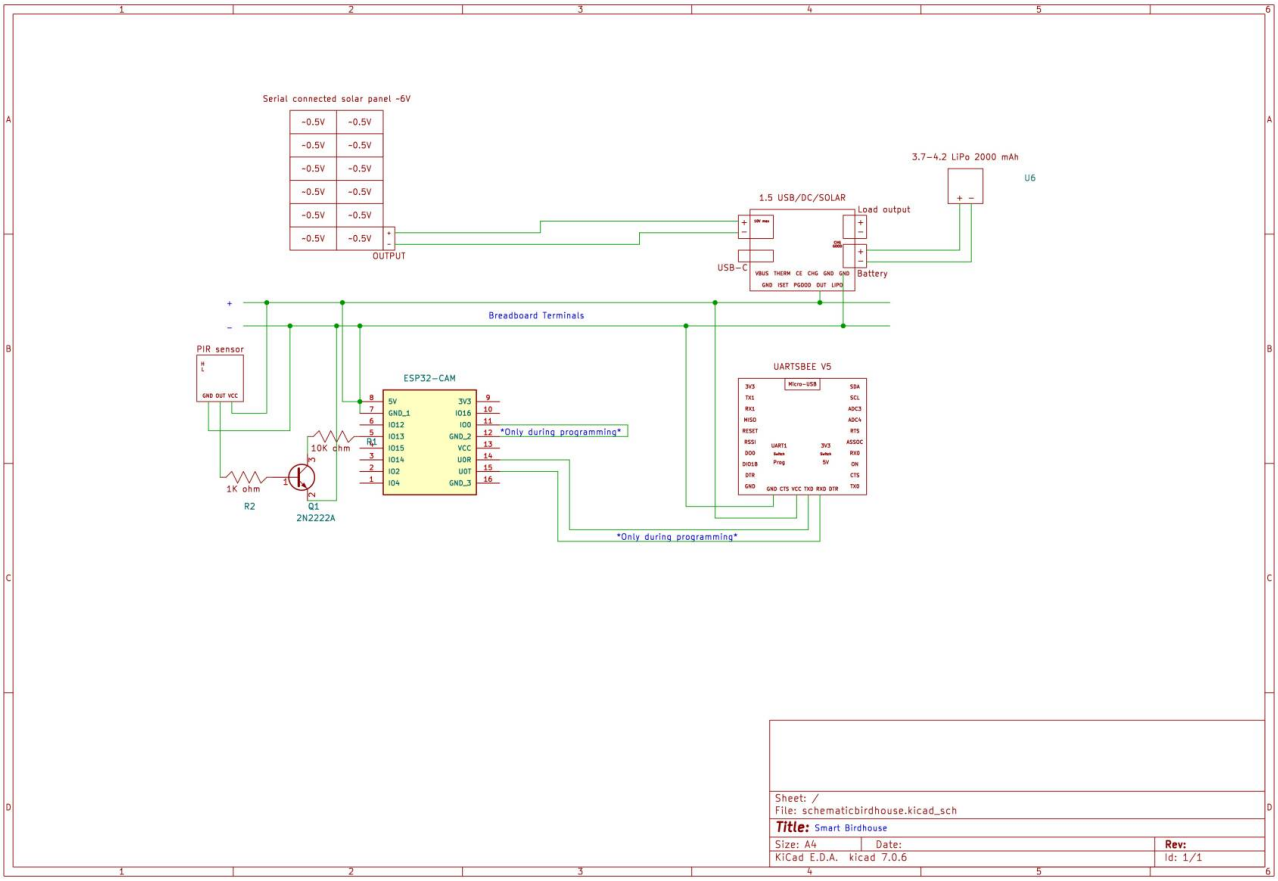


Fig 8. System schematic made with KiCad

3.0 Results (osama)

3.1 Arduino IDE C++ code(Osama)

This code is adapted from Random Nerd Tutorials: (13)

```
#include "esp_camera.h"
#include "Arduino.h"
#include "FS.h"                // SD Card ESP32
#include "SD_MMC.h"            // SD Card ESP32
#include "soc/soc.h"           // Disable brownour problems
#include "soc/rtc_cntl_reg.h"  // Disable brownour problems
#include "driver/rtc_io.h"
#include <EEPROM.h>            // read and write from flash memory

// define the number of bytes you want to access
#define EEPROM_SIZE 1

// Pin definition for CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

int pictureNumber = 0;

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detector

  Serial.begin(115200);
  //Serial.setDebugOutput(true);
  //Serial.println();
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
```

```
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA; // FRAMESIZE_ +
QVGA|CIF|VGA|SVGA|XGA|SXGA|UXGA
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }
  // Init Camera
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  //Serial.println("Starting SD Card");
  if(!SD_MMC.begin()){
    Serial.println("SD Card Mount Failed");
    return;
  }
  uint8_t cardType = SD_MMC.cardType();
  if(cardType == CARD_NONE){
    Serial.println("No SD Card attached");
    return;
  }
  camera_fb_t * fb = NULL;
  // Take Picture with Camera
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    return;
  }
```

```
  // initialize EEPROM with predefined size
  EEPROM.begin(EEPROM_SIZE);
  pictureNumber = EEPROM.read(0) + 1;
  // Path where new picture will be saved in SD Card
  String path = "/picture" + String(pictureNumber) +".jpg";
  fs::FS &fs = SD_MMC;
  Serial.printf("Picture file name: %s\n", path.c_str());
  File file = fs.open(path.c_str(), FILE_WRITE);
  if(!file){
    Serial.println("Failed to open file in writing mode");
  }
  else {
    file.write(fb->buf, fb->len); // payload (image), payload length
    Serial.printf("Saved file to path: %s\n", path.c_str());
    EEPROM.write(0, pictureNumber);
    EEPROM.commit();
  }
  file.close();
  esp_camera_fb_return(fb);
  // Turns off the ESP32-CAM white on-board LED (flash) connected to GPIO 4
  pinMode(4, OUTPUT);
  digitalWrite(4, LOW);
  rtc_gpio_hold_en(GPIO_NUM_4);
  delay(2000);
  Serial.println("Going to sleep now");
  delay(2000);
  esp_deep_sleep_start();
  Serial.println("This will never be printed"); }
```

The initial expriment ran for 1 week and did not attract any birds in fact, therefore a better location was chosen: GFL garden

The bird photos collected will be reviewed, and a conclusion about the bird species biodiversity in Kamp-Lintofrt, GFL timeline of experiment: 1 month

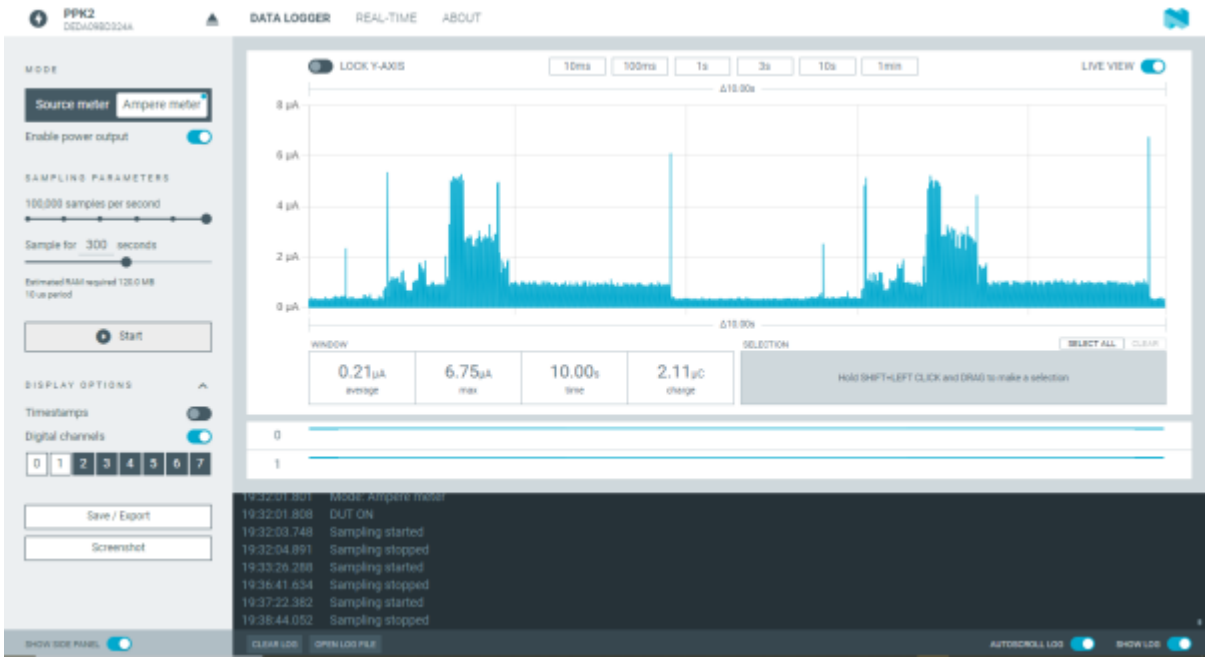
4.0 Discussion (osama)

4.1 Design considerations, Power Profiler II Nordic + power saving considerations (henrydon)

The system should be optimized to consume the least power possible. This can be achieved by programming the ESP32-CAM to switch to several sleep modes, such as in the table:

as noted from the table, wireless communication requires a lot of power

By using the power profiler tool, the power consumption of the system can be examined to compare modem sleep and deep sleep:



Battery run time

by using a tool to calculate how long the provided battery can provide power to the whole system, all while being charged by a 6V 1A solar panel

This can also be estimated manually:

Generally, if a 5V battery has a 1 Ah capacity (or 1000 mAh), then it theoretically powers a 1 A consumer for 1 h based on the formula:

Charge capacity = discharge time x charge consumption + the power formula P = V x I,

If 2000 mAh battery discharges into a system that consumes ~10 microamperes, and triggers 10 times day, each trigger consumes an average of 220 milliamperes for 6 seconds:

220 milli amperes x 6 seconds x 10 times a day = 13,200 mcoloumb or 13.2 columbs in a day when the system is triggered (this is of course increased if wireless communication is activated to send data

to a server, for example.

There are 86,400 seconds in a day, 60 of them are considered for operation = 86,340 seconds, multiplied by 10 microamperes = 863.4 milliamperes or 0.8634 amperes

average daily current discharge = 13.2 + 0.8634 = 14 amperes

average daily power consumption = battery operating voltage x average daily current consumption = 4.2 v x 14 = 58.8

to fully discharge a 2 Ah LiPo battery, assuming it is not charged with a solar panel, find the time t = battery capacity/average current consumption = 2 A x 3600 / 14A = 514.2 h or 21.4 days

For a more optimal design, some tips:

In this case, birdfood is placed outside in front of the birdhouse, and the camera is placed facing the outside. Ensure that the PIR sensor is protected one way or another, since it is imperative for monitoring purposes

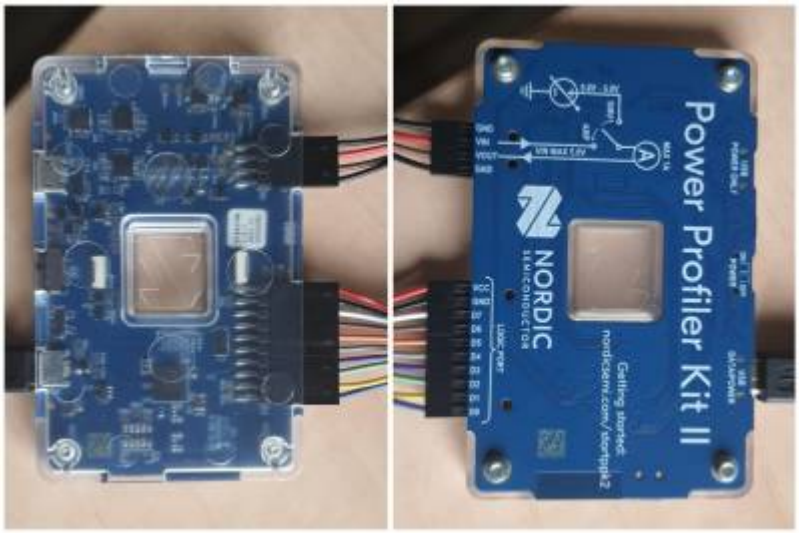
when possible, design a pcb board for your system to make it more compact

when possible, take as many precautions to protect the birdhouse from damage or decay, caused by humidity or raifall. install a cover around the rims on the top so that the water can runoff (also for the sake of shading the birds) , drill or cut a hole in the bottom part and create a gradient or incline from the inside, to drain out any water inflitration. Seal any openings with waterproof tape or glue (between the solar panel roof and the top hatch)

temperature and humidity sensors can also be added on to the system, which can highlight behaviors and preferred conditions for certain avian species

3.2 Python script to transfer photos (Osama)

(14)



5.0 Conclusion (Ismail)

To sum up the project and the effort that was put into it, we did face several challenges while developing a fully functioning bird house with integrated bird vision. For example, none of the group members had gained any experience with the utilized components in the past. Thus, it was quite overwhelming when we started working on the project. The number of components required for this project was relatively high; we used an ESP32-CAM along with an SD card, a FT232R UARTUartSBee V5 (FTDI), a solar module, a PIR module, and a battery along with a charge regulator. Each of these components had rather unique properties, which in turn intensified the challenge of getting them all to work simultaneously. However, with enough dedication, we managed to program and connect all components in the best feasible way. The knowledge gained from previous modules in the field of physics and programming greatly aided us while accomplishing this project.

6.0 References

1. Allan, B. M. et al. (2018). Futurecasting ecological research: The rise of technoecology. Ecosphere, 9(5), e02163. https://doi.org/10.1002/ecs2.2163

2. https://www.nabu.de/tiere-und-pflanzen/voegel/gefaehrdungen/24661.html

3.https://ec.europa.eu/eurostat/web/products-eurostat-news/-/edn-20220514-1

4. https://www.coastalwiki.org/wiki/Measurements_of_biodiversity#Biodiversity_indices

5. https://docs.ai-thinker.com/en/esp32-cam

6. FT232R USB UART IC Datasheet Version 2.16: https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf (Accessed on 22/07/2023)

7. https://www.pveducation.org/pvcdrom/solar-cell-operation/short-circuit-current

8. https://www.pveducation.org/pvcdrom/solar-cell-operation/open-circuit-voltage

9. PIR module (infrared module): https://components101.com/sensors/hc-sr501-pir-sensor#:~:text=The%20PIR%20sensor%20stands%20for,in%20the%20pin%20diagram%20above.

10. transistor function

11. Battery: https://www.adafruit.com/product/2011

12. Battery LiPo charger: https://www.adafruit.com/product/4755

13. Random nerd

14. python script for server upload

15. deep sleep