Step-by-step Instructions on How to Build: The SMALL Device

The Bird Feeder Collective
Alexandros Pallaris, Oskar Bolinder, Simon Ljungbeck

I. Introduction

THE SMALL (Self-Maintained Aviator Lunch Locale) device is a data-logging, internet-connected bird feeder. It contains sensors intended to detect a bird when it lands on the feeding pirch, and measure the weight of the remaining food. A small computer contained in the top section controls these sensors, and through the antenna on the top sends this data to the internet, where it can be viewed from anywhere.

The SMALL device was originally thought up as a way to modify a pre-existing bird feeder; it became its own device to simplify the design and manufacturing process. Producing your own version of the SMALL device will take a combination of constructing (or modifying) a base bird feeder, mounting some electronics including the antenna to it, and then running the software needed for it to function. Simple modifications could be made to our device, to expand or simplify the design, but we will refrain from describing more or less than the way that we built ours. In this document we first give a brief description of how the SMALL device works, and the different parts that comprise it, then we describe the steps one can use to replicate our design, accompanied by images to show the various steps.

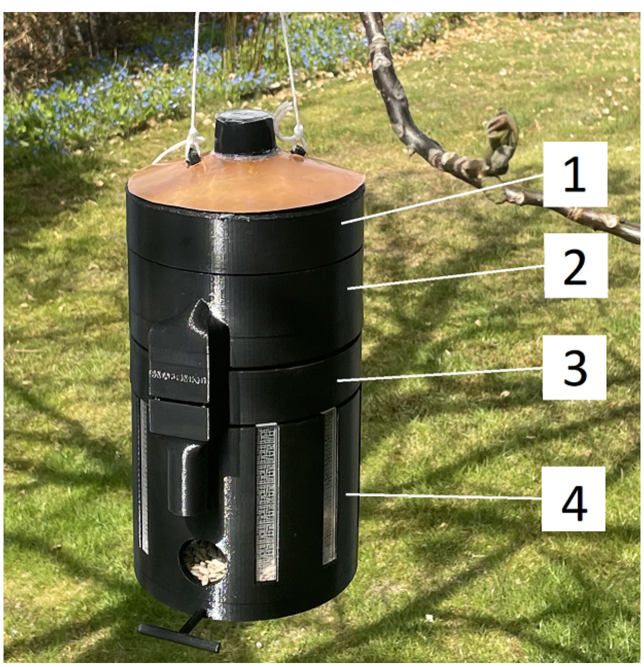

Fig. 1. The SMALL device, with labels showing the top section (**1**), the middle section (**2**), the lower section (**3**), and the bottom section (**4**).

A. Structure

In this subsection we describe the fully functioning final design (pictured in Fig. 1), the parts the make it up, and how it works. More detailed description of each part can be found by the step describing its construction in section II.

The bird feeder is conical in shape, with four different sections (see Fig. 1 and 3). On top is the roof from which the feeder is held by a branch or such, consisting of two tying points, a metal sheet, and the antenna. Our middle section is attached below and houses the electronics. Inside, the weight sensor, or load cell, is screwed into the middle from below, so that the weight of everything underneath the weight sensor can be measured. The lower section is then screwed into the bottom of the weight sensor, and holds the PIR sensor in a tube pointed toward the front of the feeding hole. This tube serves as a blinder so that the PIR sensor will only detect the presence of a bird if it is directly before the feeding hole. Finally, the bottom section is attached below the lower section, consisting of a hollow cylinder which holds the bird food, a pirch for birds to land on as they take food, and a hole by which the birds access the food.

In our system, the bird feeder works together with a separate receiver device that the bird feeder sends data to. This receiver is placed close to a WiFi router, and relays the received data to the web server. Fig. 2 shows a picture of the receiver.

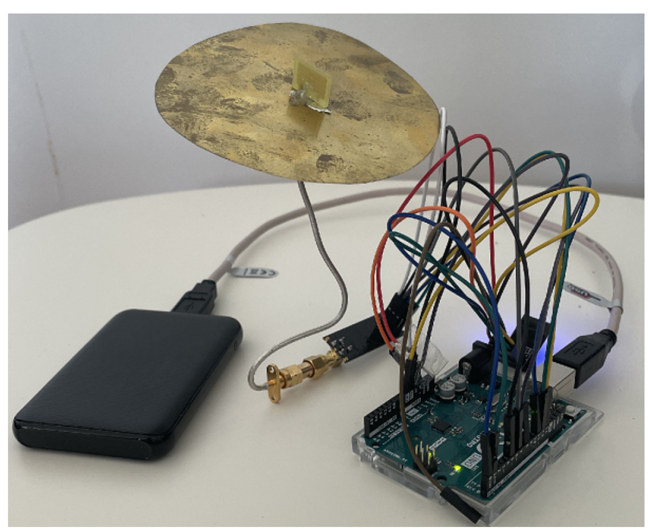

Fig. 2. The receiver device, plugged into a power bank.

Fig. 3. The sections of the SMALL device, showing them separately.

B. Electronic components

All of the electronics, including the PIR sensor, the weight sensor, and the antenna, are connected to the middle section of the device, which houses the Arduino, the HX711 ADC (HX) to the weight sensor, and the NRF24l01+ transceiver module (NRF) which the antenna is connected to. The two sensors connect to the Arduino, which relays the data from the sensors through the NRF to the antenna. From here the information is sent to the receiver which consists of another Arduino with a second antenna, and an ESP8266 WiFi module capable of connecting to a router. This is necessary because the NRF, a low-power and high-range module, cannot directly connect with WiFi. The receiver uses WiFi to upload the data to the web server where the information is stored, analyzed, and plotted on the web page to show you the history of the bird feeder.

A few things happen in the process from the sensor being activated to the data being stored on the web page:

- First, the weight sensor reports a weight (this happens periodically) or the PIR sensor detects motion
- The Arduino receives the detection or weight, and prepares the message to send to the web server
- The data is sent through the NRF module to the transmitting antenna, which relays the message to the receiving antenna
- The receiver, which is connected to the internet, relays the message to the web server
- Finally, the server stores the data and displays plots on the web page

II. Instructions

A. Structure

A good place to start is the chassis of the bird feeder: a 3-D printed multi-part plastic shell, designed to hold the electronics, the bird food, and to be hung from a tree as in Fig. 1.

The structure of SMALL device was designed mainly in the CAD program Fusion 360, in four separate parts which were separately 3-D printed and then connected. Once the 3-D model was complete (files can be found in Appendix A) the parts were exported from the CAD program as .stl files, to be prepared for 3-D printing. We had access to a Prusa printer with PETG filament, so to prepare the files we used the program PrusaSlicer which prepared the files for the specific printer and filament that we had. Part of this preparation involved the automatic adding of structural support so that each part could be printed easily. Fig. 4 shows how the model with structural support looks like. We then exported the files from PrusaSlicer to a memory card, and from the memory card to the 3-D printers to begin printing our parts. Each part took from around 12-26 hours to print.

Once printed, we carefully removed the structural support using screwdrivers, a knife, and ultimately our bare hands, taking extra care to avoid damaging the fragile sections which connect the different parts of the SMALL device. The top part should be able to hold the middle part, and the lower part the bottom.

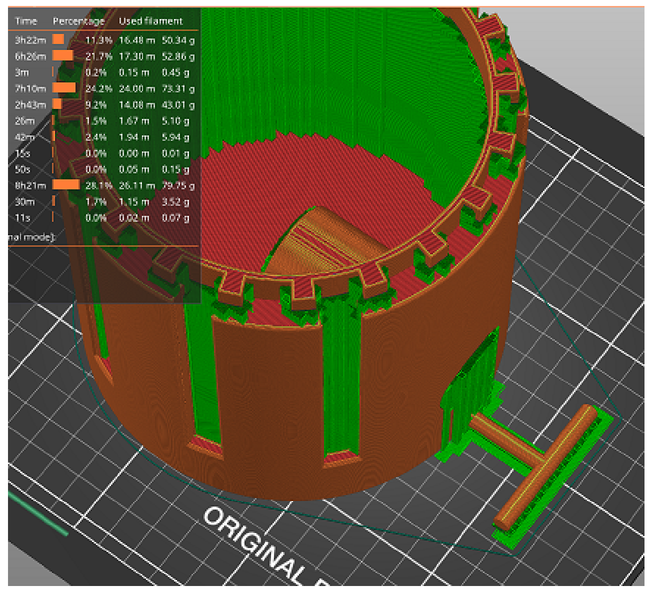

Fig. 4. A section of the bird feeder (orange/red) with structural support (green) as shown by PrusaSlicer.

B. Electronics

Having the bird feeder chassis, the next step is to set up the electronics. The system consisted of the following electronic components:

- 2x Arduino UNO-SMD-R3
- PIR sensor WPSE314
- Weight sensor HX711 Load Cell Amplifier
- 2x nRF2401 Single Chip 2.4 GHz Radio Transciever
- ESP8266-01 WiFi module

The subsections below describe how to connect each component to the full system.

1) PIR Sensor: We used the PIR sensor WPSE314 from Whadda in this project (see Fig. 5). The sensor has three pins that should be connected to the Arduino in the bird feeder with PCB connection cables. Table I shows how the connections are made up.

There are two screws on the PIR sensor, one to adjust the sensitivity and another for the reset delay which controls

TABLE I
CONNECTIONS BETWEEN PIR SENSOR AND ARDUINO.

PIR sensor	Arduino UNO
VCC	*3.3V*
OUT	*Pin 3*
GND	*GND*

the shortest possible time between detections. The sensor is very sensitive, and therefore we recommend to use the lowest sensitivity. We also recommend to use the lowest reset delay, as this ranges upwards from approximately 5 seconds, and a bird is unlikely to stay for that long.

Fig. 5. The PIR sensor, capable of detecting in a wide cone out from the white bulb.

2) Weight Sensor: The weight sensor consists of two parts – a load cell and a separate analog-to-digital converter (ADC) that handles the weight data and directs it to the Arduino (see Fig. 6). The weight sensor is screwed into place between the middle and lower sections of the bird feeder, in the provided holes, and is connected to the HX711 ADC. The ADC itself then connects to the Arduino in the bird feeder. The connections can be found in table II and III.

TABLE II
CONNECTIONS BETWEEN THE LOAD CELL & HX711 ADC.

Load Cell	HX711 ADC
Green	*A+*
White	*A-*
Red	*E+*
Black	*E-*

3) nRF24L01+ Modules: We used two nRF2401+ Single Chip 2.4 GHz Radio Transciever modules from ICQUANZX in this project; one to send data from the bird feeder, and the other one to receive the data close to the WiFi router. These modules have an SMA connector, the gold-coloured part in Fig. 7, to which the antenna's coaxial cable is screwed into. Both the transmitting and the receiving radio module were connected in the same way to the Arduinos. The connections are found in table IV.

TABLE III
CONNECTIONS BETWEEN CIRCUIT CARD OF THE HX711 ADC AND THE ARDUINO.

HX711 ADC	Arduino UNO
GND	*GND*
DT	*Pin 4*
SCK	*Pin 5*
VCC	*5V*

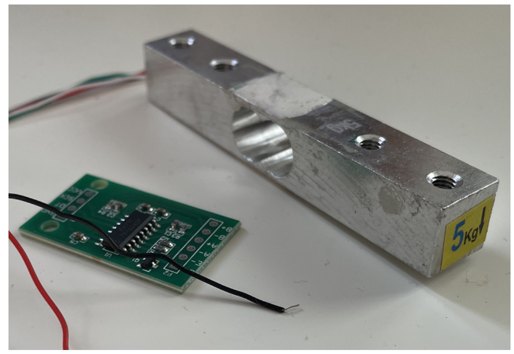

Fig. 6. The load cell and the HX711 ADC.

TABLE IV
CONNECTIONS BETWEEN AN NRF24L01+ WIFI MODULE AND AN ARDUINO UNO.

NRF24L01+	Arduino UNO
VCC	*3.3V*
GND	*GND*
CNS	*Pin 8*
CE	*Pin 7*
MOSI	*Pin 11*
SCK	*Pin 13*
MISO	*Pin 12*

The nRF24L01+ module comes with an antenna that can be seen in Fig. 7. Since we wanted to use our own antenna in this project, we simply removed the manufactured antenna from the SMA jack.

4) ESP8266 WiFi Module: The ESP8266 WiFi module (see Fig. 8) was connected to the receiving Aruduino, outside the bird feeder. Table V shows the connections.

When uploading code to the module, the *RST* pin and the *IO0* pin are used as well. Plug in the *IO0* pin to *GND* on the Arduino, and touch the *RST* pin to the *3.3V* on the Arduino just before uploading the code.

C. Code

Before putting all electronics inside the bird feeder chassis, we need to program the two Arduinos, the ESP8266 WiFi module, and the web server which displays the bird feeder data. All of this code can be found in Appendix A).

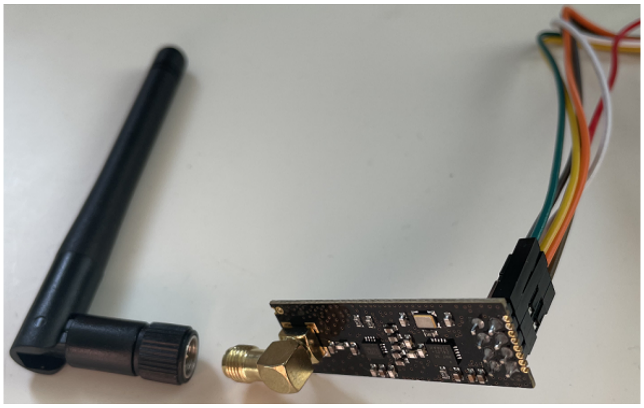

Fig. 7. The nRF24L01 module, with the included antenna shown beside it.

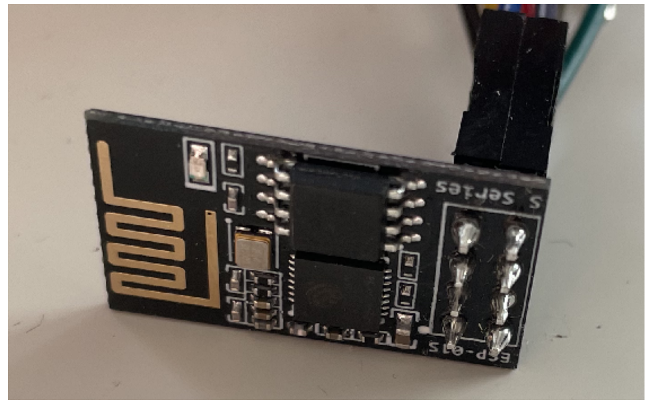

Fig. 8. The ESP8266 WiFi module.

TABLE V
CONNECTIONS BETWEEN THE ESP8266 WIFI MODULE AND THE ARDUINO UNO.

ESP8266	Arduino UNO
3.3V	*3.3V*
RX	*RX*
RST	-
IO0	-
EN	*3.3V*
IO2	*Pin 5*
GND	*GND*

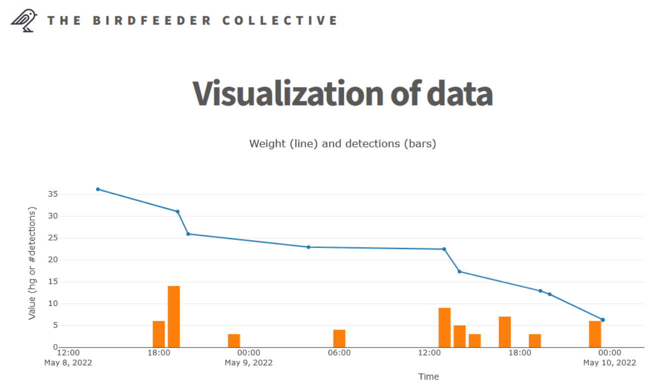

Fig. 9. The statistics page on https://birdfeedercollective.com/, showing a histogram of bird feeder detections per hour and a plot of measured weight over time.

1) Arduino and ESP8266 WiFi Module Code: The Arduinos and the ESP8266 WiFi module are most easily programmed with the Arduino IDE program in a language similar to the C language. Code is then simply uploaded to the Arduino/WiFi module via a USB cable.

2) Web Server: In this project, we bought a web server from bluehost.com and used the domain name https://birdfeedercollective.com/. If you want to have the same web page as we had (see the design in Fig. 9), you simply log in to Bluehost, go to file manager in advanced settings and put the code from Appendix A in the folder *public_html*.

D. Putting in the Electronic Parts

When everything is working on the software side, it is time to put all electronics inside the bird feeder chassis. All electronics should be placed in the middle section of the bird feeder (see Fig. 1) except the PIR sensor, which is placed in the hole in the lower section, and the weight sensor, which forms a part of the structure as it is screwed between the lower and the middle section, holding the two sections together.

With the Arduino inside the bird feeder, it is no longer possible to power it with a USB cable from the computer. Instead a power bank was used. The bird feeder has a hole in the second part designed for Powerbank 5000 from LINOCELL.

E. Antenna

At this point in the building process, there is only one step left – building our own antenna and connecting it to the system. The antenna was created from a sheet of copper-covered FR-4 substrate by using a milling machine to cut the unwanted copper away, then cut out the antenna. Fig. 10 shows the milling machine we used.

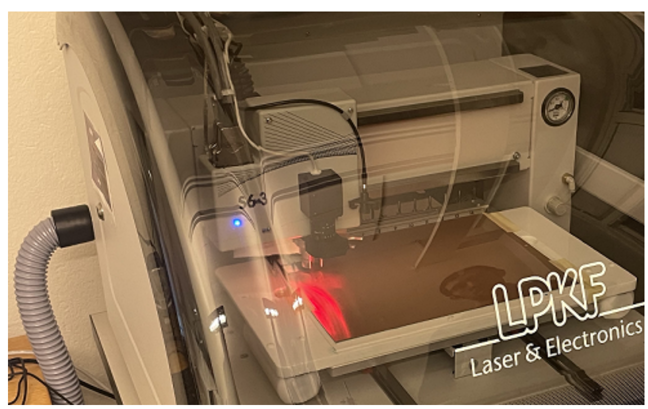

Fig. 10. The milling machine used to create the antenna.

The antenna was designed using the FEKO computational electromagnetics software. After we decided on the final design it was optimized by having the program run slight changes until it had a low reflection coefficient at the port. Once designed, the 2-D structure of the antenna was exported in a Gerber file for use with the milling machine.

F. Connecting the Antenna and Ground Plane

The ground plane is constructed from a single, thin, sheet of metal, which can be easily cut with regular scissors. To bend the plane into a cone, a single cut is made toward the centre of the sheet, which is then bent before tape or glue is applied to hold its shape. To fit over the holding points, two rectangular holes are then made in the corresponding places on the ground plane (by means of poking with a sharp object, bending the edges of the holes, and very careful use of scissors). The plane can then be fit onto the top of the feeder, where one can poke or drill a hole in its centre. It can be then be glued in place, to ensure that it doesn't shift around.

We connect the antenna to the electronics with an SMA cable (see Fig. 11), which consists of an outer conductor separated from an inner conductor. Since the antenna must be electrically connected to both the ground plane and the NRF module, we need to solder three points: the ground connection of the antenna, the centre conductor's connection to the antenna, and the outer conductor connection's to the ground plane.

Fig. 11. The soldered receiving antenna, showing the top (**left**) and bottom (**right**) sides. The ground connection of the antenna (**1**), the centre conductor's connection to the antenna (**2**), and the outer conductor connection's to the ground plane (**3**) can be seen.

The soldering (see Fig. 12) is a delicate process during which one should be careful to only form electrical connection where intended. After each connection is made, use an Ohmmeter to check that connections have been made where you want them, and not made where you do not want them. To ensure that connections are not accidentally broken, and hold everything in place, one should cover finished solders with hot glue. Glue can also be used to hold the antenna to the ground plane while you are soldering. Once finished, you can connect your antenna to a virtual network analyzer to test it. The reflection coefficient should be lower than -10 dB over the useful frequency band, from about 2.4 to 2.5 GHz. However, as long as it is lower than -10 dB over a range of even just 10 MHz that it shares with the receiver, that is fine as you can control the transmitting frequency within 1 MHz.

G. Finalize the Device

At this stage, there are only some small details to fix before the bird feeder can be hung on a tree. First, we need some windows for the third section of the bird feeder. Those were cut out from a Plexiglass sheet with a laser cutter (see Fig. 13). For aesthetic reasons, we decided to also burn a pattern into the windows. The pattern we used can be found in the instruction file for the laser cutter in Appendix A.

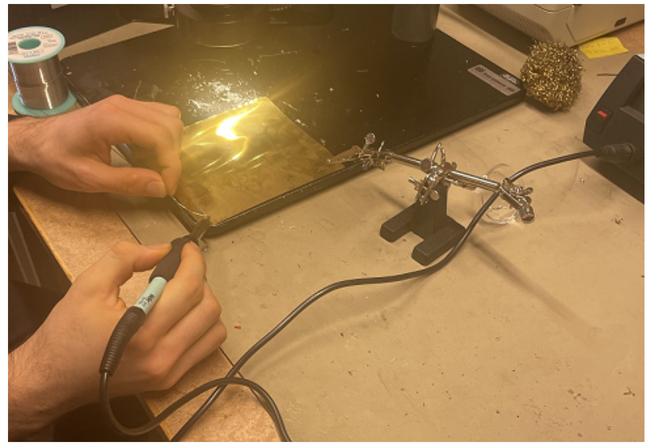

Fig. 12. Soldering the antenna to the ground plane.

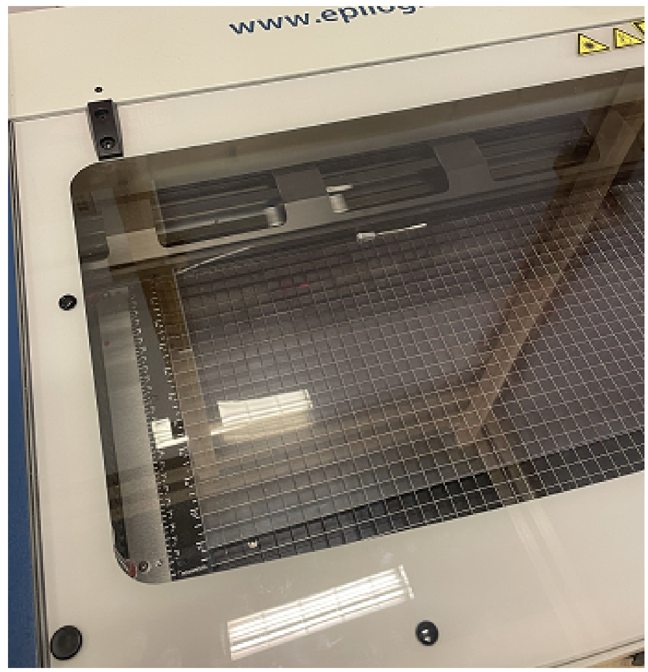

Fig. 13. The laser cutter, just before cutting out our windows.

The windows can be simply inserted into the bottom section, and all of the electronics placed in their sections. Then, all that remains is to fill the bird feeder with bird food, attach some ropes to the top part, and hang it in a tree.

Appendix A
Supplementary Files

The code and files used for our device can be found at: https://github.com/SL2000s/birdfeeder_pub. This page contains the 3-D models of the bird feeder structure, the antenna CAD file and structure, and the code used with the electronics.

Acknowledgment

The authors would like to thank our mentors Johan Lundgren and Mats Gustafsson for their advice and criticism throughout the project. We would also like to thank Martin Nilsson for help manufacturing the antennas.