Bucknell UNIVERSITY

Sending data from ESP32 to Notehub, via Notecard

Guide to set up Notehub and transmit data from an ESP32 to Notehub, via Notecard

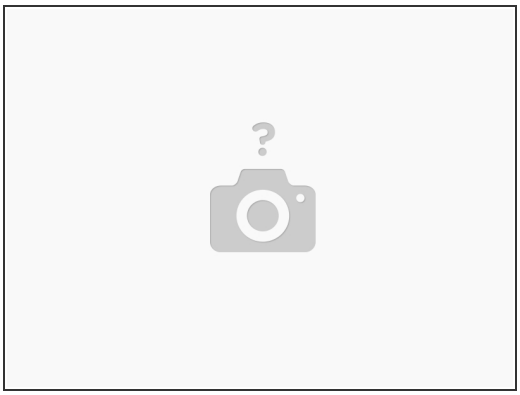
Written By: Asta Rustad

blues inc

INTRODUCTION

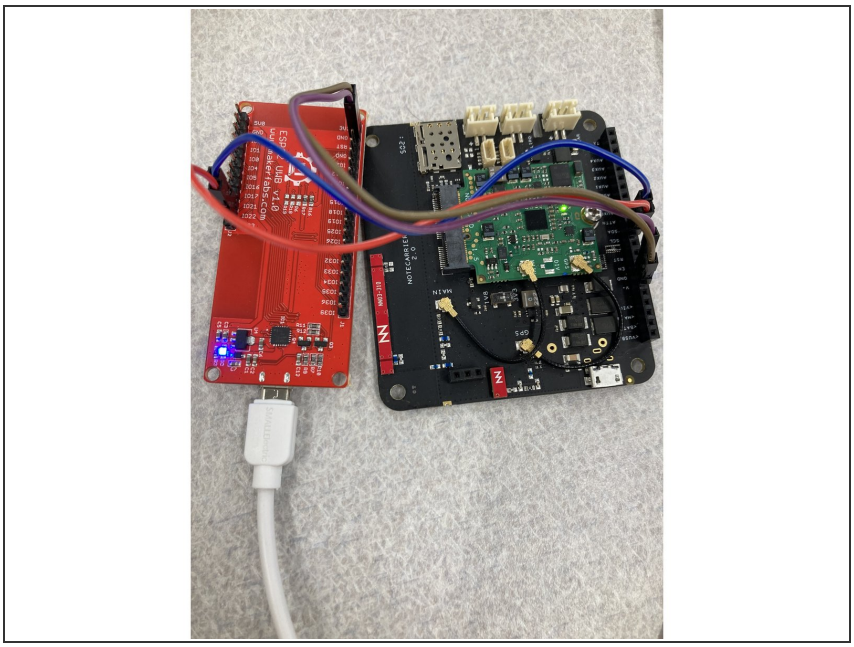
This guide shows you how to transmit data from a ESP32 UWB, via a Notecard, and to Notehub. It also shows how to create a project in Notehub, such that the transmitted data can be viewed.

Step 1 — Bill of Materials



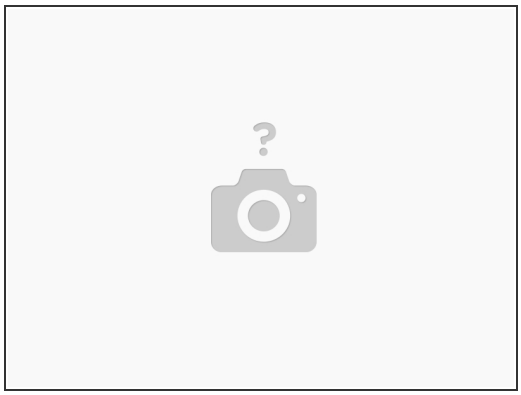
- ESP32 UWB
- Notecarrier-A
- USB Type A to USB Micro B
- 4 male to female wires

Step 2 — Setting up Hardware



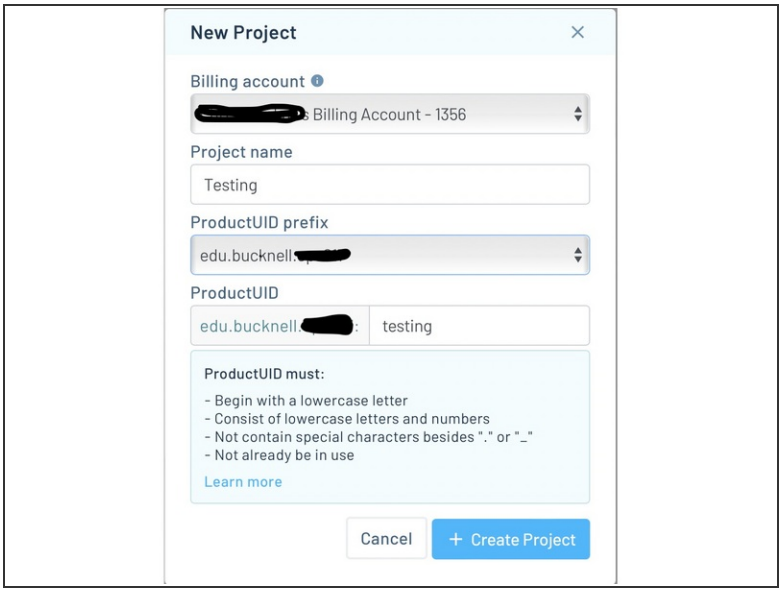
- Connect the following pins:
- 3V3 on ESP32 to V+ on Notecarrier
- GND (the GND next to the 3V3 pin is the one we used) on ESP32 to GND on Notecarrier
- IO21 on ESP32 to SDA on Notecarrier
- IO22 on ESP32 to SCL on Notecarrier
- Connect the ESP32 UWB to your computer using USB

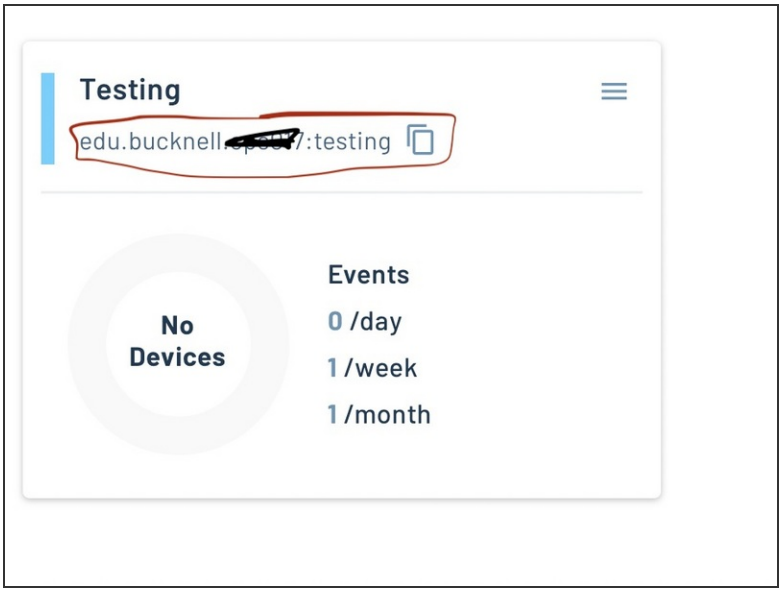
Step 3 — Setting up Arduino



- Download Arduino IDE : https://www.arduino.cc/en/software
- Once downloaded, choose the correct board:
- Navigate to Tools->Board->Boards Manager
- Search “esp32” and click “Install”
- Once installed (can take a few minutes), navigate to Tools->Board->esp32 and select “ESP32 Wrover Module”
- Select the port by navigating to Tools->Port and select the port you want to use. In my case it was called “/dev/cu.usbserial-02310491”
- Install the Notecard library: https://dev.blues.io/tools-and-sdks/libr...

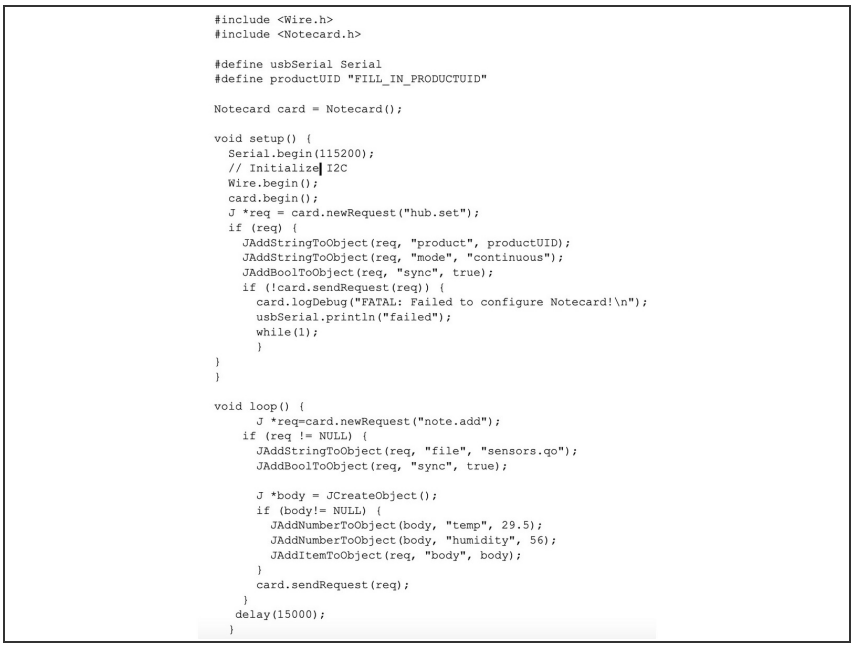
Step 4 — Setting up Notehub





- Each Notecard comes with 5000 “consumption credits.” To utilize these credits, you need to create a Notehub account. After setting up a Notehub account, you can create a project. Then, by adding the ProjectUID to your code, the data sent from the Notecard will appear in your Project.
- Sign up to Notehub: https://notehub.io/sign-up
- Create a project by clicking “Create Project”. Something similar to image 1 should pop up.
- Add a project name and complete the ProductUID. Click "Create Project."
- The ProductUID is circled in red in the second image. Copy this ProductUID.

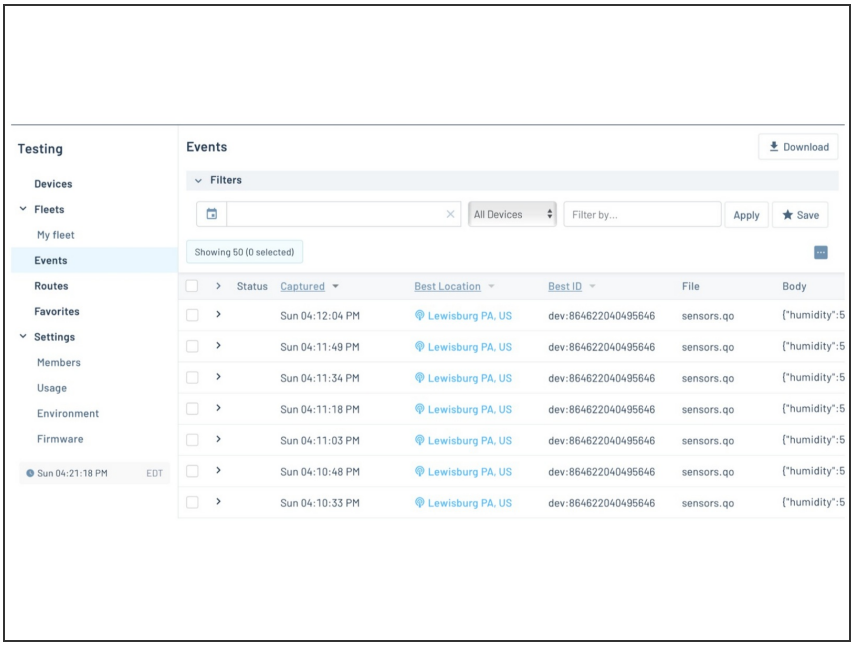
Step 5 — Sending Data to Notehub



```
#include <Wire.h>
#include <Notecard.h>

#define usbSerial Serial
#define productUID "FILL_IN_PRODUCTUID"

Notecard card = Notecard();

void setup() {
  Serial.begin(115200);
  // Initialize I2C
  Wire.begin();
  card.begin();
  J *req = card.newRequest("hub.set");
  if (req) {
    JAddStringToObject(req, "product", productUID);
    JAddStringToObject(req, "mode", "continuous");
    JAddBoolToObject(req, "sync", true);
    if (!card.sendRequest(req)) {
      card.logDebug("FATAL: Failed to configure Notecard!\n");
      usbSerial.println("failed");
      while(1);
      }
}
}

void loop() {
      J *req=card.newRequest("note.add");
    if (req != NULL) {
      JAddStringToObject(req, "file", "sensors.qo");
      JAddBoolToObject(req, "sync", true);

      J *body = JCreateObject();
      if (body!= NULL) {
        JAddNumberToObject(body, "temp", 29.5);
        JAddNumberToObject(body, "humidity", 56);
        JAddItemToObject(req, "body", body);
      }
      card.sendRequest(req);
    }
   delay(15000);
  }
```

- Copy the code from the image into Arduino IDE. The code is a basic “Hello World,” which sends a hardcoded temperature and humidity value.
- Under #define productUID "FILL_IN_PRODUCTUID", add the ProductUID from Notehub. It should look something like #define productUID "edu.bucknell.XXX:testing"
- Compile and upload the code to the ESP32 by clicking the "->" icon in Arduino IDE

Step 6 — Viewing Uploaded Data



- In Notehub, click on the project you have created.
- Select "Events"
- Data should now have been sent to Notehub, and it should look something like the image.

