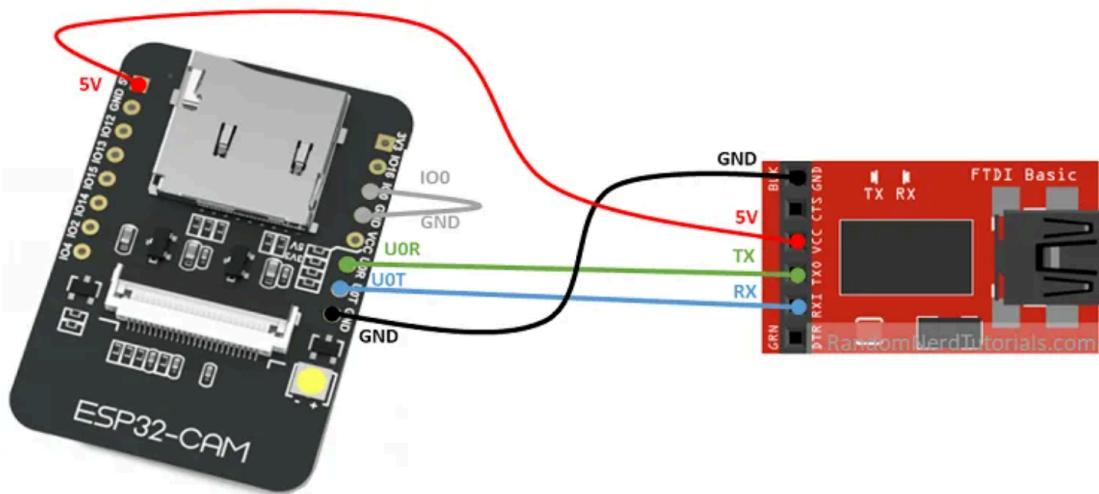


CAPSTONE WASTE CAN DEVELOPMENT

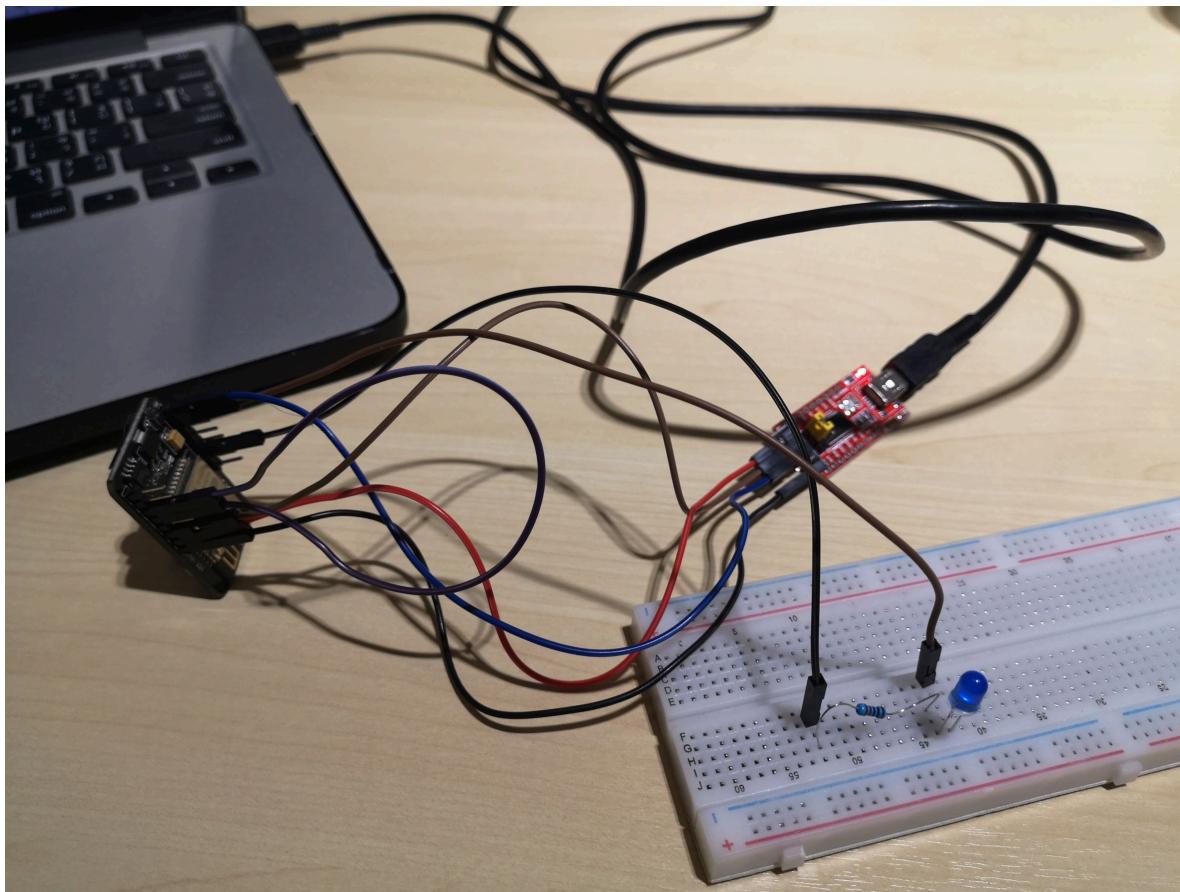
JUNE 24 (DID NOT WORK)

Establishing Serial Communication Between ESP32 CAM and Arduino Through Blinking Lights

1. Uploaded code to ESP 32 CAM through FTDI programmer using the basic schematic below:



+ added **GND** (ESP32 CAM) to negative pin of LED and **GPIO 14** (ESP32 CAM) to positive pin of LED (this schematic worked when only blinking the lights)



Code on ESP32 CAM:

The screenshot shows the Arduino IDE interface with the title bar "ESP_Serial | Arduino 1.8.10". The main window displays the code for "ESP_Serial". The code uses SoftwareSerial to read data from pin 3 and write to pin 1. It includes a setup function to initialize the serial port at 9600 baud and set pin 14 as an output. The loop function reads data from the serial port, checks if it's 526, and if so, toggles an LED connected to pin 14. Otherwise, it turns off the LED. After each toggle, it waits for 500 milliseconds. The status bar at the bottom shows "Done uploading.", "Leaving...", and "Hard resetting via RTS pin...".

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial (3,1); //Rx, Tx pins
int data;
const int ledPin = 14;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);

}

void loop() {
  data = Serial.read();
  delay(1000);

  if(data == 526){
    digitalWrite (ledPin, HIGH); // turn on the LED
    delay(500); // wait for half a second or 500 milliseconds
    digitalWrite (ledPin, LOW); // turn off the LED
    delay(500); // wait for half a second or 500 milliseconds
  }
  else{
    digitalWrite(ledPin, LOW);
  }
}


```

2. Uploaded code to Arduino Uno



Code on Arduino Uno:

The screenshot shows the Arduino IDE interface with the title bar "Arduino_Serial | Arduino 1.8.10". The code editor contains the following sketch:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.write(526);
  delay(1000);
}
```

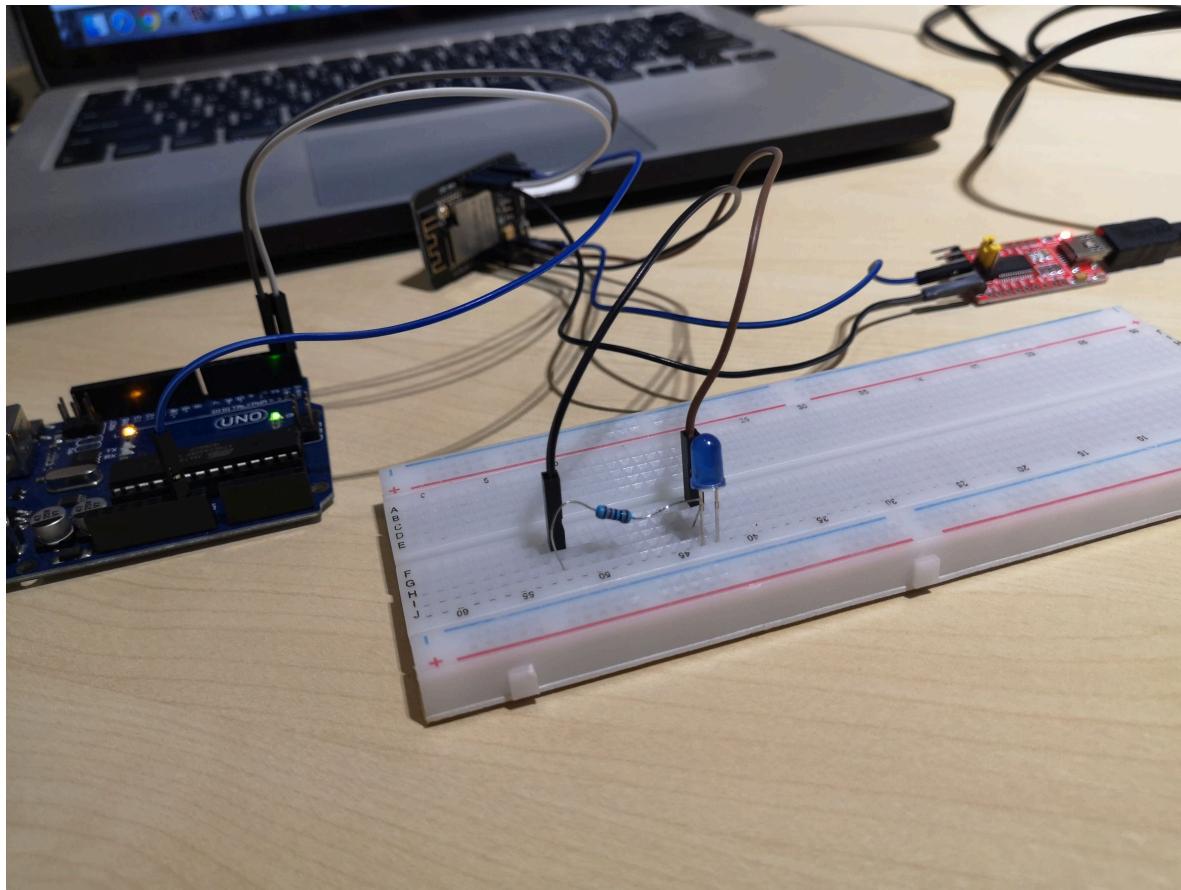
3. On ESP32 CAM, I did this to run code:

6) When you see the "**Done uploading**" message, you need to remove **GPIO 0** from **GND** and press the RST button to run your new code.

4. After that, I disconnected TX and RX pins from FTDI programmer and connected them to Arduino Uno instead (TX Arduino - RX ESP32, RX

Arduino - TX ESP32)

5. Connected GND of ESP32 to GND of Arduino Uno



Result: the light turned on for around 1 second, then remained off throughout. Even after clicking reset on ESP32 CAM again, it only turned on for 1 second and remained off throughout.