

```
// Adafruit FONA and Arduino Code
```

```
// Libraries
```

```
#include <Adafruit_SleepyDog.h>
```

```
#include "Adafruit_FONA.h"
```

```
#include <SoftwareSerial.h>
```

```
// LED pin
```

```
const int ledPin = 13;
```

```
// Pins
```

```
#define FONA_RX 2
```

```
#define FONA_TX 3
```

```
#define FONA_RST 4
```

```
// Buffer
```

```
char replybuffer[255];
```

```
// Instances
```

```
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
```

```
SoftwareSerial *fonaSerial = &fonaSS;
```

```
// Fona instance
```

```
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
```

```
uint8_t type;
```

```
// Thing name
```

```
String yourThing1 = "board2server";
```

```
String yourThing2 = "server2board";
```

```
// RESET THESE: Flagging and String Checking
```

```
String incoming_stream = "";
```

```
bool gprs_flag = false;
```

```
bool dweet_flag = false;
```

```
void setup() {
```

```
  pinMode(13, OUTPUT);
```

```
  // Initial serial
```

```
  while (!Serial);
```

```
  Serial.begin(115200);
```

```
  Serial.println(F("FONA reading SMS"));
```

```
  Serial.println(F("Initializing....(May take 3 seconds)"));
```

```

fonaSerial->begin(4800);
if (! fona.begin(*fonaSerial)) {
    Serial.println(F("Couldn't find FONA"));
    while (1);
}

type = fona.type();
Serial.println(F("FONA is OK"));
Serial.print(F("Found "));
switch (type) {
    case FONA800L:
        Serial.println(F("FONA 800L")); break;
    case FONA800H:
        Serial.println(F("FONA 800H")); break;
    case FONA808_V1:
        Serial.println(F("FONA 808 (v1)")); break;
    case FONA808_V2:
        Serial.println(F("FONA 808 (v2)")); break;
    case FONA3G_A:
        Serial.println(F("FONA 3G (American)")); break;
    case FONA3G_E:
        Serial.println(F("FONA 3G (European)")); break;
    default:
        Serial.println(F("???")); break;
}

// Print module IMEI number.
char imei[15] = {0}; // MUST use a 16 character buffer for IMEI!
uint8_t imeiLen = fona.getIMEI(imei);
if (imeiLen > 0) {
    Serial.print("Module IMEI: "); Serial.println(imei);
}

// Setup GPRS settings
//fona.setGPRSNetworkSettings(F("internet"));
while (!gprs_flag)
{
    fona.setGPRSNetworkSettings(F("wholesale"), F(""), F(""));
    delay (15000);
    if (!fona.enableGPRS(true))
    {
        Serial.println(F("GPRS re-establishing connection"));
    }
    else
    {

```

```

    Serial.println(" ^____^");
    Serial.println("(° ㄣ °)");
    Serial.println("c      つ");
    Serial.println("(つ /");
    Serial.println("(ノ");
    Serial.println("GPRS connection successful");
    gprs_flag = true;
}
}
delay(2000);
}

void loop() {
    // Loop
    // Prepare request
    uint16_t statuscode;
    int16_t length;
    String url = "http://dweet.io/get/latest/dweet/for/board2server_";
    char buf[80];
    url.toCharArray(buf, url.length());

    Serial.print("Request: ");
    Serial.println(buf);

    // Send URL to Dweet.io
    while (!dweet_flag)
    {
        if (!fona.HTTP_GET_start(buf, &statuscode, (uint16_t *)&length))
        {
            Serial.println("Dweet reattempt");
            dweet_flag = false;
            delay(2000);
        }
        else
        {
            dweet_flag = true;
        }
    }
    while (length > 0) {
        while (fona.available()) {
            char c = fona.read();
            incoming_stream += (char)c;
            // Serial.write is too slow, we'll write directly to Serial register!
#ifdef __AVR_ATmega328P__ || defined(__AVR_ATmega168__)
            loop_until_bit_is_set(UCSR0A, UDRE0); /* Wait until data register empty.*/
#endif
        }
    }
}

```

```

        UDR0 = c;
    #else
        Serial.write(c);
    #endif
    length--;
}
}
fona.HTTP_GET_end();

// PIN CONTROL
if (incoming_stream.indexOf("\led_1\":"on\\") >= 0)
{
    Serial.println("LED: ON");
    digitalWrite(13, HIGH);
}
else if (incoming_stream.indexOf("\led_1\":"off\\") >= 0)
{
    Serial.println("LED: OFF");
    digitalWrite(13, LOW);
}
else
{
    Serial.println("LED: RESET");
    digitalWrite(13, LOW);
}

// Reset flags for next run
gprs_flag = false;
dweet_flag = false;
incoming_stream = "";
delay(3000);
}

```