

```
/******
```

This is an example for our Adafruit FONA Cellular Module

Designed specifically to work with the Adafruit FONA

----> <http://www.adafruit.com/products/1946>

----> <http://www.adafruit.com/products/1963>

----> <http://www.adafruit.com/products/2468>

----> <http://www.adafruit.com/products/2542>

These cellular modules use TTL Serial to communicate, 2 pins are required to interface

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.

BSD license, all text above must be included in any redistribution

```
*****/
```

```
/*
```

THIS CODE IS STILL IN PROGRESS!

Open up the serial console on the Arduino at 115200 baud to interact with FONA

Note that if you need to set a GPRS APN, username, and password scroll down to the commented section below at the end of the setup() function.

```
*/
```

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

```
#define FONA_RX 9
```

```
#define FONA_TX 8
```

```
#define FONA_RST 4
```

```
#define FONA_RI 7
```

```
// this is a large buffer for replies
```

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

```
unsigned long StartTime = millis();
```

```
// We default to using software serial. If you want to use hardware serial
```

```
// (because softserial isnt supported) comment out the following three lines
```

```
// and uncomment the HardwareSerial line
```

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

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

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

```

// Hardware serial is also possible!
// HardwareSerial *fonaSerial = &Serial1;

// Use this for FONA 800 and 808s
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
// Use this one for FONA 3G
//Adafruit_FONA_3G fona = Adafruit_FONA_3G(FONA_RST);

//uint8_t readline(char *buff, uint8_t maxbuff, uint16_t timeout = 0);

uint8_t type;

void setup() {
  //while (!Serial);

  Serial.begin(115200);
  Serial.println(F("FONA basic test"));
  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[16] = {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);
}

// Optionally configure a GPRS APN, username, and password.
// You might need to do this to access your network's GPRS/data
// network. Contact your provider for the exact APN, username,
// and password values. Username and password are optional and
// can be removed, but APN is required.
//fona.setGPRSNetworkSettings(F("your APN"), F("your username"), F("your
password"));
fona.setGPRSNetworkSettings(F("hologram"));

// Optionally configure HTTP gets to follow redirects over SSL.
// Default is not to follow SSL redirects, however if you uncomment
// the following line then redirects over SSL will be followed.
//fona.setHTTPSRedirect(true);

while(fona.getNetworkStatus() !=1 && fona.getNetworkStatus() != 5) {
  // read the network/cellular status
  uint8_t n = fona.getNetworkStatus();
  Serial.print(F("Network status "));
  Serial.print(n);
  Serial.print(F(": "));
  if (n == 0) Serial.println(F("Not registered"));
  if (n == 1) Serial.println(F("Registered (home)"));
  if (n == 2) Serial.println(F("Not registered (searching)"));
  if (n == 3) Serial.println(F("Denied"));
  if (n == 4) Serial.println(F("Unknown"));
  if (n == 5) Serial.println(F("Registered roaming"));
  Serial.println("Waiting 5 seconds...");
  delay(5000); // wait 5 seconds
}
unsigned long CurrentTime = millis();
unsigned long ElapsedTime = CurrentTime - StartTime;
Serial.println("Connected to Hologram Cellular...");
Serial.println(ElapsedTime);

//*** Time ***/
// enable network time sync
Serial.println(F("Enable network time sync"));

```

```
fona.enableNetworkTimeSync(true);
```

```
// read the time
```

```
char buffer[23];
```

```
fona.getTime(buffer, 23); // make sure replybuffer is at least 23 bytes!
```

```
Serial.print(F("Time = ")); Serial.println(buffer);
```

```
}
```

```
void loop() {
```

```
}
```