

```

//*****
// 24.7.16
// Tests new old phone using FONA 3G & Arduino
// Operates very basic functions of a phone
// Gets number from dial and connects
// Hangs up when cradle switch is pressed
// Thanks to Adafruit for FONA library and other coding
//*****

#include "Adafruit_FONA.h"

#define FONA_RX 2
#define FONA_TX 3
#define FONA_RST 4

// this is a large buffer for replies
char replybuffer[255];
//software serial
#include <SoftwareSerial.h>
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;

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

Adafruit_FONA_3G fona = Adafruit_FONA_3G(FONA_RST);

uint8_t type;
char tele_num_str1[14]; // storage for dialed number
byte digit = 0;
byte count = 0;
boolean calling = false;
//*****

void setup() {

  pinMode(8, INPUT_PULLUP); //end of pulse train switch, HIGH at end
  pinMode(9, INPUT_PULLUP); //pulse train switch - pulse is HIGH
  pinMode(10, INPUT_PULLUP); //handset cradle switch LOW handset OFF cradle
  pinMode(11, INPUT_PULLUP); //call switch LOW when pressed

  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 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);
  }

  fona.setGPRSNetworkSettings(F("giffgaff.com"), F("giffgaff"), F(""));
}
//*****
void flushSerial() {
  while (Serial.available())
    Serial.read();
}
//*****

```

```

void loop() {

  for(byte x = 0; x < count; x++){          //erase any existing number
    tele_num_str1[x] = '\0';
  }
  count = 0;
  while(digitalRead(10) == LOW){          // handset off cradle

    while(digitalRead(8) == LOW){          // detect current pulse train being generated
      delay(20);                          // allow for bounce
      if(digitalRead(9) == HIGH){          // register a pulse
        digit++;
        while(digitalRead(9) == HIGH){    // wait for end of current pulse
          delay(20);                      // allow for bounce
        }
      }
    }
  }

  if(digit != 0){                          //10 pulses represents zero
    if(digit == 10){
      digit = 0;
    }

    tele_num_str1[count] = digit + '0';    // change digit to character

    digit = 0;
    count++;
    if(count > 12){                        // limit no of digits to 12
      count = 12;
    }
  }

  if(digitalRead(11) == LOW){              // call switch pressed

    tele_num_str1[count + 1] = '\0';
    Serial.print(tele_num_str1);
    Serial.println();

    break;                                  //break out of while loop
  }
}

  if(digitalRead(10) == LOW){              //handset still off cradle
    calling = true;
    flushSerial();
    Serial.print(F("Call #"));
    Serial.println();
    Serial.print(F("Calling "));
    Serial.println(tele_num_str1);
    if (!fona.callPhone(tele_num_str1)) {
      Serial.println(F("Failed"));
    } else {
      Serial.println(F("Sent!"));
    }
  }

  while(digitalRead(10) == LOW){           //wait for cradle switch to be pressed to hang up
  }
  delay(50);

  if(calling){                             // now hang up
    if (! fona.hangUp()) {
      Serial.println(F("Failed"));
    } else {
      Serial.println(F("HANG UP OK!"));
    }
    calling = false;
  }

  // flush input
  flushSerial();
  while (fona.available()) {
    Serial.write(fona.read());
  }
}

//end of main loop

```