Làm GSM mobile phone với Arduino

Tuan Nguyen 8/Jan 2019

Những component cần dùng

Mot cai GSM phone tốn khoảng 1.95tr mua tai hshop.vn, tiki va lazada.

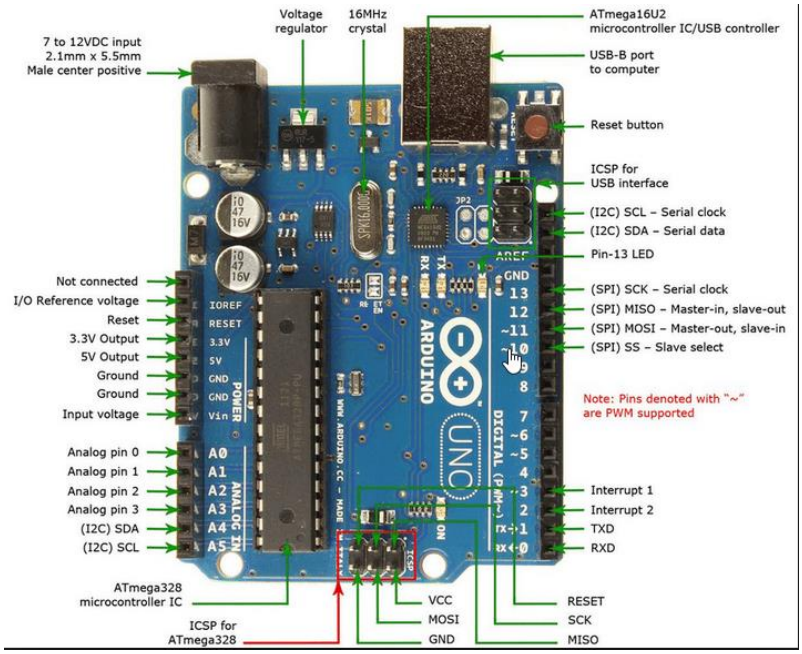
1) arduino uno + day USB: 184k (tiki - https://tiki.vn/kit-arduino-uno-r3-chip-cam-atmega328p-tang-cap-ket-noi-p7855424.html)
2) sim808 combo with GPS và cục transformer 12v: 870k (hshop; https://hshop.vn/products/arduino-gsm-gprs-gps-bluetooth-sim808-shield)
3) 3,5" HMI display (CN version) : 550K (hshop; https://hshop.vn/products/man-hinh-hmi-uart-cam-ung-dien-tro-nextion-3-5-inch)

NB: tiếng trung quoc hơi khó xài, bạn nên mua Nextion 3.2, UK version ; như thế ban phải resize nhung tấm hình về 240x400 ; https://hshop.vn/products/man-hinh-hmi-uart-cam-ung-dien-tro-nextion-3-2-inch

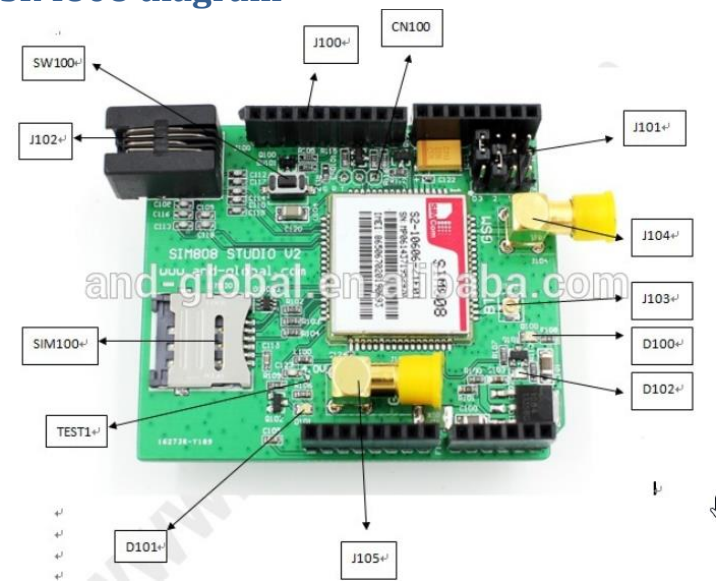
4) Test/jumper wires : 22K (hshop; https://hshop.vn/products/day-camtest-board-duc-duc-nhieu-koch-co)
5) USB UART (for testing; windows 10) : 55k (hshop; https://hshop.vn/products/mach-chuyen-usb-uart-cp2102)
6) headphone/mic (4 pin RJ11) : 150k (lazada; https://www.lazada.vn/-i252834972-s337116630.html?urlFlag=true&mp=1?)

Total: ~2tr VND

Arduino diagram



SIM808 diagram



Install Arduino IDE

https://www.arduino.cc/en/Main/Software

Click Windows version

Install GSM GPRS GPS library

Click “clone or download” -> download zip (bên tay phải) https://github.com/MarcoMartines/GSM-GPRS-GPS-Shield

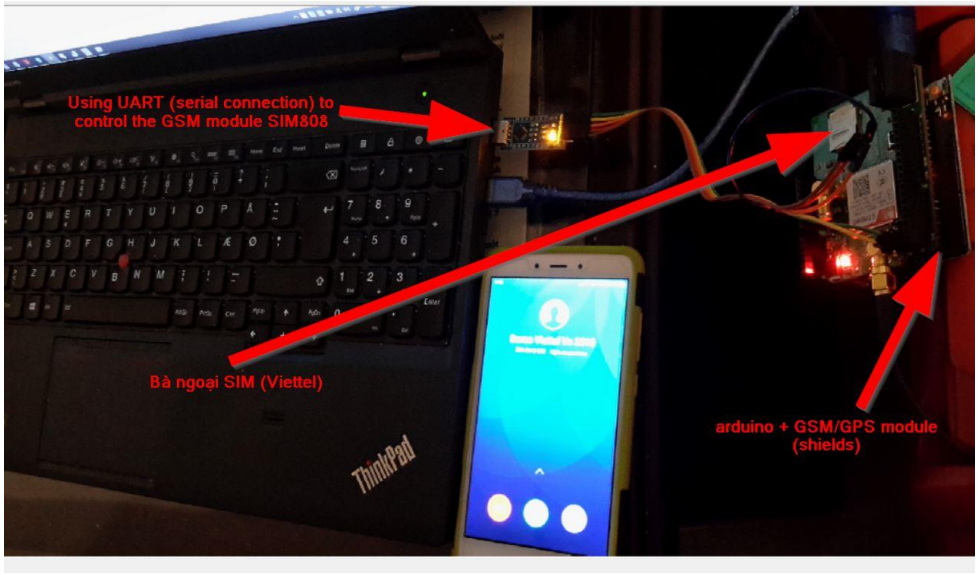
Mở zip file, copy cái thư viện GSM-GPRS-GPS-Shield-GSMSHIELD vào c:\Program Files (x86)\Arduino\libraries\

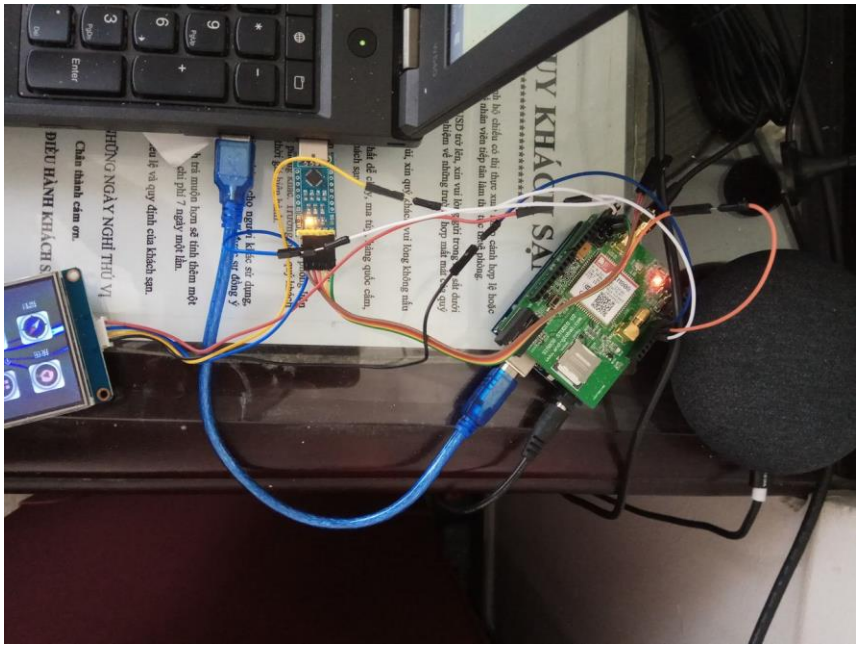
Đóng và mở lại arduino editor.

Test SIM808

Trước khi xài nên test cái arduino + sim808 với “usb-uart-cp2102”

Cận thận lấy sim808 nối vào bên trên con arduino, xem hình dưới đây. Nếu thac mac thì hỏi oncle Google để xem nhung tấm hình khác rõ hơn.





Mở arduino và test

~~$0D$0A = return~~
~~AT$0D$0A~~
call:
ATD0366291927~~;$0D$0A~~
hang up: ATH~~$0D$0A~~

NB: $0D$0A chỉ quan trong khi bạn xài 1 cai serial terminal khác như “hercules_3-2-8 serial editor”

SMS:
AT+CMGF=1$0D$0A
AT+CMGS="0366290000"$0D$0A
>hello
>test
ctrl+z
DONE

HMI display

Chú ý, có 2 version, cho CN (Hoa) và UK (English). Nếu mua CN HMI bạn không thể nào xài cái "HMI editor" của UK được!! Khi mua ở hshop.vn họ có viết kỹ, nhớ đọc và download software từ link o hshop.vn

CN: "USART HMI.exe"
UK: "NextionEditor.exe"

Nếu xài CN version, hãy mở UK version bên cạnh để mà mò được

Những cái object (như hình, text, button, hotspot, scrolling text..) của HMI mỗi cái có 1 ID và 1 cái name riêng, khi bạn activate (ấn nó trên HMI display) thì cái ID nó chạy vào Arduino và điều trị nó theo cái program của bạn, thí du:

a) NexHotspot h2 = NexHotspot(0, 9, "m2"); //hotspot added HANG UP
Cái hotspot "m2" này có ID=9 ở page 0.

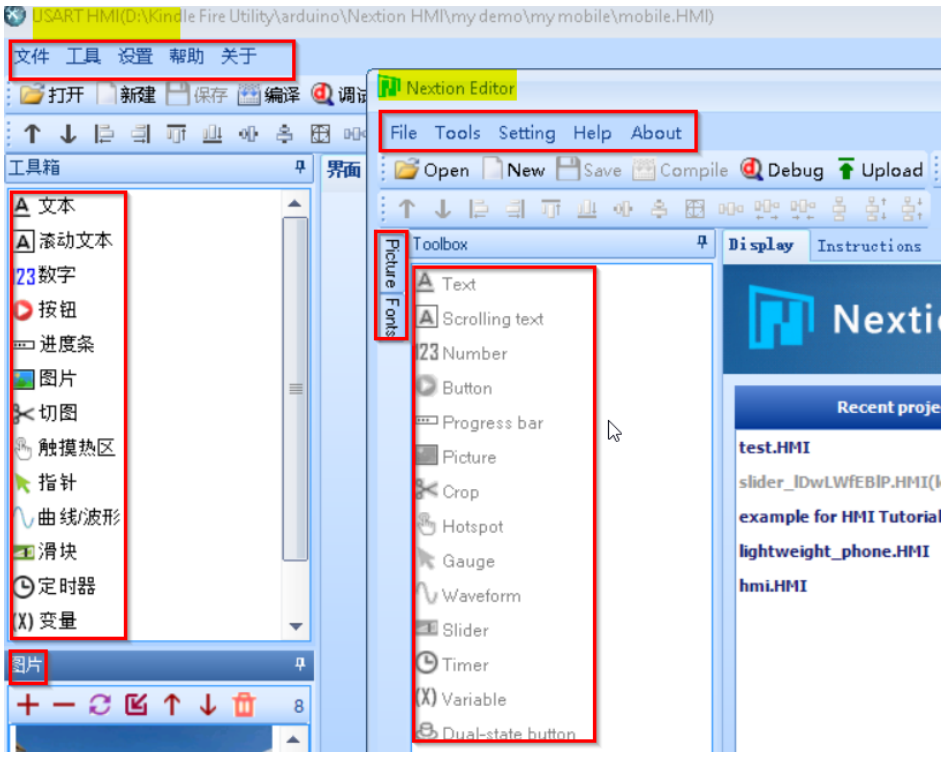
b) NexTouch *nex_events_call[] = {&h2} //register cái event hotspot h2

c) Định nghĩa function for h2, cho nó cái tên h2PushCallback

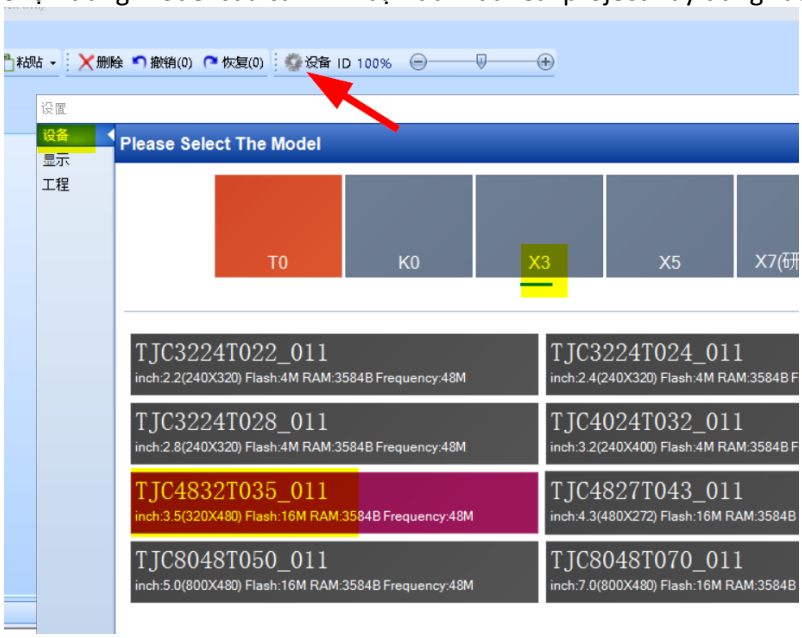
```
void h2PushCallback(void *ptr)  // Press event for button hotspot h2  HANG UP
{
  GSMSerial.println(F("ATH"));
  Serial.println(F("Hangup Call"));
  smartDelay(1000);
}
```

d) Khi bạn ấn h2 nó sẽ activate cai function h2PushCallback

```
  h2.attachPush(h2PushCallback);  // Button press ; hangup
```



Chọn đúng model của cái HMI bạn đã mua. Cái project này dùng TJC4832T035_011 (3.5 inch CN model)

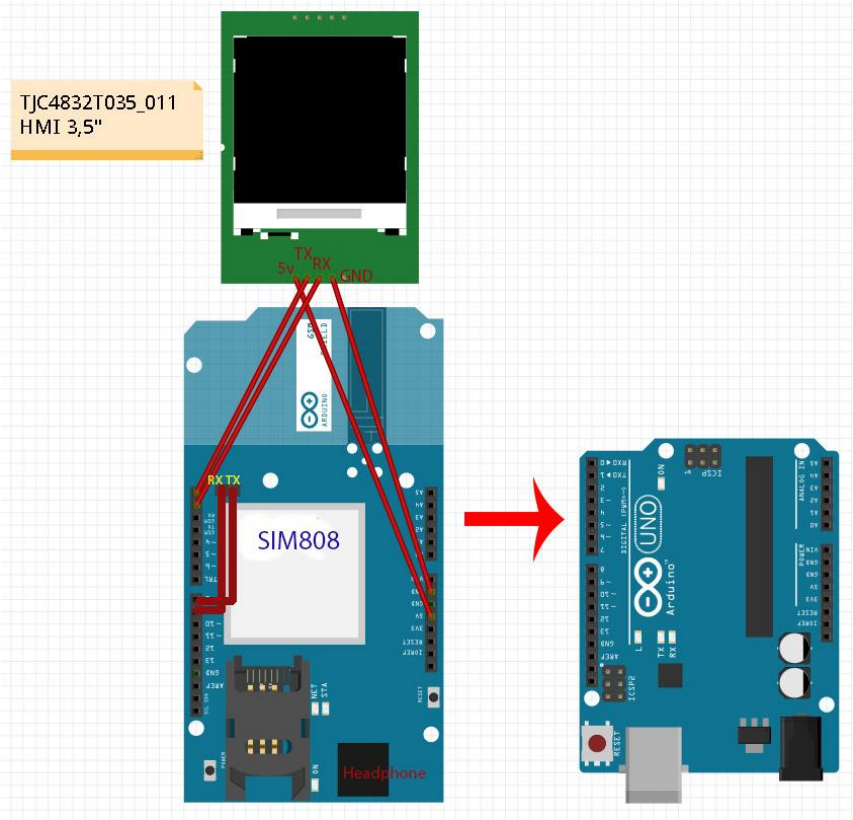


Info:
TJC clone of NX (Itead Nextion models)

https://ckblog2016.wordpress.com/tag/tjc/

- Displays called TJC ... are being marketed by Taojingchi (TJC) for the Chinese market.
- Display labeled NX ... are distributed by ITEAD Studios for the international market.
- Both companies are based in Shenzen.
- TJC and Nextion work together to make sure the firmware from one company is not compatible with the other .
- Both companies have their own editor, which differs slightly in functionality.
- It is to use the Nextion editor for GUI development for NX displays and the USART HMI Editor with Chinese labeling for the TJC displays. A mix does not work!

The phone



The Code

gsmphone.ino

```
/*
Made by Tuan Nguyen
Last update: 07/02/2019
# for Uno arduino - need to edit d:\Program Files (x86)\Arduino\libraries\ITEADLIB_Arduino_Nextion-master\NexConfig.h
according to Nextion hmi library guide (see github below)
a) Only work IF HMI connect to the HW serial pin 1,2 (RX,TX), NOT the softwareserial. The softwareserial is for GSM
module

b) about different libraries
*  puts the critical TX/RX device (GPRS) on the super-efficient hardware serial port (i.e., Serial);
*  puts the most TX-intensive device (screen) on the next best software serial port (i.e., AltSoftSerial); and
*  puts the least-used TX device (MP3) player on the least-efficient serial port (i.e., NeoSWSerial).

c) due to only 2kB ram, dont use ModuleSerialGps for GPS, only AT command, it works see d:\Kindle Fire
Utility\arduino\gps\gps.ino
d) Do you mean that I cannot use GPS functionality or turn on GPS if the GSM network is not registered? Then it makes
sense because it seems to not able to register to GSM network. Both the GSM and GPS have antenna.
e) saving RAM (2K)
macro F in print to avoid using RAM
https://www.baldengineer.com/arduino-f-macro.html
Don't wait until your code starts acting weird. Whenever you use strings in a print() method, get into the habit of
wrapping the string with a F() macro. This way you don't have to worry about RAM getting wasted on something that can't
be changed anyway.
Notice:This microcontroller only offers 2,048 bytes of RAM. 2k, that's it

Sketch uses 17184 bytes (53%) of program storage space. Maximum is 32256 bytes.
Global variables use 1424 bytes (69%) of dynamic memory, leaving 624 bytes for local variables. Maximum is 2048 bytes.

7 feb 2019
Sketch uses 11466 bytes (35%) of program storage space. Maximum is 32256 bytes.
Global variables use 1466 bytes (71%) of dynamic memory, leaving 582 bytes for local variables. Maximum is 2048 bytes.

enable 4 more buttons, still work!!
```

```
10/feb gprs
need to use delay instead smartdelay, something with serial and interups...otherwise dont get any data from gprs
NOT WORK really, unstable, just read a small file 34bytes hello.txt
NEED TO USE CASE AT d:\Kindle Fire Utility\arduino\gprs2\gprs2.ino to read big file.

Sketch uses 11746 bytes (36%) of program storage space. Maximum is 32256 bytes.
Global variables use 1545 bytes (75%) of dynamic memory, leaving 503 bytes for local variables. Maximum is 2048 bytes.
Low memory available, stability problems may occur.

Sketch uses 11744 bytes (36%) of program storage space. Maximum is 32256 bytes.
Global variables use 1531 bytes (74%) of dynamic memory, leaving 517 bytes for local variables. Maximum is 2048 bytes.
Sketch uses 12678 bytes (39%) of program storage space. Maximum is 32256 bytes.
Global variables use 1529 bytes (74%) of dynamic memory, leaving 519 bytes for local variables. Maximum is 2048 bytes.


*/

// #include "Arduino.h"
#include "Nextion.h"   // Include the nextion library (the official one)
https://github.com/itead/ITEADLIB_Arduino_Nextion
#include "AltSoftSerial.h"  // replace softserial library
//#include <ModuleSerialGsm_Sms.h>  //use it for received sms

AltSoftSerial GSMSerial(8, 9);  //rx,tx ;  for GSM sim808 shielder
AltSoftSerial gprs(8, 9);
String str,str3,Number,Key,SMSTxt;


//NexHotspot h0 = NexHotspot(0, 7, "m0");  //hotspot added   SMS
//NexHotspot h1 = NexHotspot(0, 8, "m1");  //hotspot added    CALL
NexHotspot h2 = NexHotspot(0, 9, "m2");  //hotspot added    HANG UP
NexHotspot h3 = NexHotspot(0, 12, "m3");  //hotspot added  answer
NexHotspot h4 = NexHotspot(0, 13, "m4");  //hotspot added   GPS
//NexText t0 = NexText(0, 6, "t0");  // Text box added, so we can read it
NexSlider g0 = NexSlider(0, 5, "g0");  // Slider added
NexButton b0 = NexButton(1, 2, "b0");  // Button added nr 1
NexButton b1 = NexButton(1, 3, "b1");  // Button added nr 2
NexButton b2 = NexButton(1, 4, "b2");  // Button added nr 3
NexButton b3 = NexButton(1, 5, "b3");  // Button added nr 4
NexButton b4 = NexButton(1, 6, "b4");  // Button added nr 5
NexButton b5 = NexButton(1, 7, "b5");  // Button added nr 6
NexButton b6 = NexButton(1, 8, "b6");  // Button added nr 7
NexButton b7 = NexButton(1, 9, "b7");  // Button added nr 8
NexButton b8 = NexButton(1, 10, "b8");  // Button added nr 9
//NexButton b9 = NexButton(1, 11, "b9");  // Button added *
NexButton b10 = NexButton(1, 12, "b10");  //  nr 0
NexButton b11 = NexButton(1, 13, "b11");  // #
NexButton b12 = NexButton(1, 17, "b12");  // CLEAR button
NexButton b13 = NexButton(1, 18, "b13");  // pre-text "on my way"
//NexButton b14 = NexButton(1, 19, "b14");  // Button added #
//NexButton b15 = NexButton(1, 20, "b15");  // Button added #
//NexButton b16 = NexButton(1, 21, "b16");  // Button added #
//NexButton b17 = NexButton(1, 22, "b17");  // Button added ;  Back button
NexHotspot h5 = NexHotspot(1, 23, "m0");  //hotspot added   CALL
NexHotspot h6 = NexHotspot(1, 24, "m1");  //hotspot added    SMS
NexText t1 = NexText(1, 15, "t1");  // Text box added, so we can read it
NexHotspot h7 = NexHotspot(0, 15, "m5");  //hotspot added   quotes

NexTouch *nex_events_call[] = {&h2,&h3,&h4,&b0,&b1,&b2,&b3,&b4,&b5,&b6,&b7,&b8,&b10,&b11,&b12,&b13,&h5,&h6,&h7,NULL}; //
using in loop ()
// NexTouch *nex_events_call[] =
{&h0,&h1,&h2,&h3,&h4,&b0,&b1,&b2,&b3,&b4,&b5,&b6,&b7,&b8,&b9,&b10,&b11,&b12,&b13,&b14,&b15,&b16,&b17,&h5,&h6,NULL}; //
using in loop ()


void h2PushCallback(void *ptr)  // Press event for button hotspot h2  HANG UP
{
  GSMSerial.println(F("ATH"));
  Serial.println(F("Hangup Call"));
  smartDelay(1000);
}

void h3PushCallback(void *ptr)  // Press event for button hotspot h3  Answer phone
{
  GSMSerial.println(F("ATA"));
  Serial.println(F("Answer Call"));
  smartDelay(1000);
}

void h4PushCallback(void *ptr)  // Press event for button hotspot h4  GPS position
```

```
{
  GSMSerial.println("AT+CGPSINF=0");
  String str2 = GSMSerial.readString();
  //str = str2.substring(46,66);
//1046.158400,10642.149700,9.800000,20190207120309
  str = str2.substring(106,154);
  Serial.println(str);
//output of the GPS should be placed at loop, if you put here, nothing happens
//7 feb 2019: not true, I need to have it here also before it works (yes 2 places) !!
  PushNex(str);
  smartDelay(1000);

}


//PAGE2

void h5PushCallback(void *ptr)  // Press event for button hotspot h5 CALL
{
  String Nr="ATD"+Number + ";";
  digitalWrite(13, HIGH);  // Turn ON internal LED
  //GSMSerial.println(F("ATD0366291927;")); // ATDxxxxxxxxxx; -- watch out here for semicolon at the end!!
  GSMSerial.println(Nr); // ATDxxxxxxxxxx; -- watch out here for semicolon at the end!!
  Serial.println(F("Calling  ")); // print response over serial port
  Serial.println(Nr);
  smartDelay(1000);
  digitalWrite(13, LOW);
}  // End of press event

void h6PushCallback(void *ptr)  // Press event for button hotspot h6 ; send SMS
{
  String Nr2="AT+CMGS=\""+Number + "\"\r";
  digitalWrite(13, HIGH);
  GSMSerial.println(F("AT+CMGF=1"));    //Sets the GSM Module in Text Mode
  smartDelay(1000);  // Delay of 1000 milli seconds or 1 second
  GSMSerial.println(Nr2); // Replace x with mobile number
  //GSMSerial.println(F("AT+CMGS=\"0366291927\"\r")); // Replace x with mobile number
  smartDelay(1000);
  GSMSerial.println(SMSTxt);// The SMS text you want to send
  //GSMSerial.println(F("sim808 SMS TUAN GSM TESTING....."));// The SMS text you want to send
  smartDelay(100);
  GSMSerial.println((char)26);// ASCII code of CTRL+Z
  smartDelay(1000);
  digitalWrite(13, LOW);
}  // End of press event



void b0PushCallback(void *ptr)  // Press event for button nr 1
{
 Key = "1";
 Number += Key;
 Serial.println(Number);
}  // End of press event


void b1PushCallback(void *ptr)  // Press event for button nr 2
{
 Key = "2";
 Number += Key;
 Serial.println(Number);
}  // End of press event

void b2PushCallback(void *ptr)  // Press event for button nr 3
{
 Key = "3";
 Number += Key;
 Serial.println(Number);
}  // End of press event

void b3PushCallback(void *ptr)  // Press event for button nr 1
{
 Key = "4";
 Number += Key;
 Serial.println(Number);
}  // End of press event

void b4PushCallback(void *ptr)  // Press event for button nr 5
{
 Key = "5";
 Number += Key;
```

```
 Serial.println(Number);
}  // End of press event

void b5PushCallback(void *ptr)  // Press event for button nr 6
{
 Key = "6";
 Number += Key;
 Serial.println(Number);
}  // End of press event

void b6PushCallback(void *ptr)  // Press event for button nr 7
{
 Key = "7";
 Number += Key;
 Serial.println(Number);
}  // End of press event

void b7PushCallback(void *ptr)  // Press event for button nr 8
{
 Key = "8";
 Number += Key;
 Serial.println(Number);
}  // End of press event

void b8PushCallback(void *ptr)  // Press event for button nr 9
{
 Key = "9";
 Number += Key;
 Serial.println(Number);
}  // End of press event


void b10PushCallback(void *ptr)  // Press event for button nr 0
{
 Key = "0";
 Number += Key;
 Serial.println(Number);
}  // End of press event

void b11PushCallback(void *ptr)  // Press event for button #
{
 Key = "#";
 Number += Key;
 Serial.println(Number);
}  // End of press event


void b12PushCallback(void *ptr)  // Press event for button CLEAR
{
 Number = "";
 Serial.println(Number);

}

void b13PushCallback(void *ptr)  // Press event for pre-text
{
 SMSTxt = "On my way";
 Serial.println(SMSTxt);
}

/*
void b14PushCallback(void *ptr)  // Press event for pre-text
{
 SMSTxt = "Cant talk";
 Serial.println(SMSTxt);
}
*/

//page 3
void h7PushCallback(void *ptr)  // Press event for button hotspot h5 quotes position, using GPRS
{
 //flush the buffer
  while(Serial.available()){Serial.read();}
  while(GSMSerial.available()){GSMSerial.read();}
  //while(gprs.available()){gprs.read();}


  gprs.println(F("AT+CMEE=1"));
  gprs.println(F("AT+CGATT=1"));
  gprs.println(F("AT+SAPBR=3,1,\"Contype\",\"GPRS\""));
```

```
  //gprs.println(F("AT+SAPBR=3,1,\"APN\",\"v-internet\""));
  //gprs.println(F("AT+SAPBR=3,1,\"USER\",\"\""));
  //gprs.println(F("AT+SAPBR=3,1,\"PWD\",\"\""));
  gprs.println(F("AT+SAPBR=3,1,\"APN\",\"v-internet\""));
  smartDelay(400);
  gprs.println(F("AT+SAPBR=1,1"));
  smartDelay(3000);
  gprs.println(F("AT+SAPBR=2,1"));

  gprs.println(F("AT+HTTPINIT\n"));
  gprs.println(F("AT+HTTPPARA=\"CID\",1"));
  gprs.println(F("AT+HTTPPARA=\"URL\",\"http://www.chezmoi.dk/hello.txt\""));
  smartDelay(1000);
  gprs.println(F("AT+HTTPACTION=0"));

  ///httpaction takes a while beforw it is ready, hence 4000 delay
  Serial.println(F("Wait 4000ms ....................."));
  smartDelay(4000);
  while(GSMSerial.available()){GSMSerial.read();}
  gprs.println(F("AT+HTTPREAD=0,60"));
  smartDelay(1500);
  readln();
     /*
    while (gprs.available()) {
    char c = gprs.read();
    Serial.println(c);}
  */
  Serial.println(F("AT+HTTPTERM"));
  delay(1000);
}

//pop text on page 1
void PushNex(String &str) {
    Serial.print("g0.txt=");  // This is sent to the nextion display to set what object name (before the dot) and what
atribute (after the dot) are you going to change.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.print(str);  // This is the text you want to send to that object and atribute mentioned before.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.write(0xff);  // We always have to send this three lines after each command sent to the nextion display.
    Serial.write(0xff);
    Serial.write(0xff);
    smartDelay(1000);
}

void PushNexP2(String &str2) {
    Serial.print("t1.txt=");  // This is sent to the nextion display to set what object name (before the dot) and what
atribute (after the dot) are you going to change.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.print(str2);  // This is the text you want to send to that object and atribute mentioned before.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.write(0xff);  // We always have to send this three lines after each command sent to the nextion display.
    Serial.write(0xff);
    Serial.write(0xff);
    smartDelay(1000);
}

void PushNexP3(String &str3) {
    Serial.print("t3.txt=");  // This is sent to the nextion display to set what object name (before the dot) and what
atribute (after the dot) are you going to change.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.print(str3);  // This is the text you want to send to that object and atribute mentioned before.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.write(0xff);  // We always have to send this three lines after each command sent to the nextion display.
    Serial.write(0xff);
    Serial.write(0xff);
    smartDelay(1000);
}

void setup() {  // Put your setup code here, to run once:
  //nexInit();
  GSMSerial.begin(9600);
  gprs.begin(9600);
  Serial.begin(9600);    // Setting the baud rate of HW Serial Monitor (Arduino)
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
```

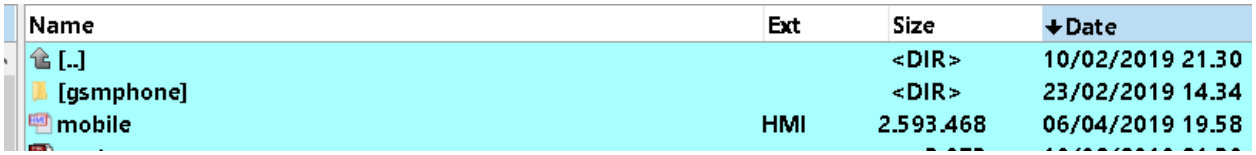
```
  }
  Serial.println(F("HELLO TO GSM PHONE"));  // activate "tools -> serial monitor" to see this
  Serial.println(F("Initializing..."));
  GSMSerial.println("AT+CGPSPWR=1");
  GSMSerial.println("AT+CGPSINF=0");

  // Format for press events: <object name>.attachPush(<object name>PushCallback);
  // Format for release events: <object name>.attachPop(<object name>PopCallback);

  h2.attachPush(h2PushCallback);  // Button press ; hangup
  h3.attachPush(h3PushCallback);  // Button press, Answer call
  h4.attachPush(h4PushCallback);  // Button press ; GPS
  h5.attachPush(h5PushCallback);  // Button press ; call
  h6.attachPush(h6PushCallback);  // Button press ; sms
  h7.attachPush(h7PushCallback);  // Button press ; quotes
  b0.attachPush(b0PushCallback);
  b1.attachPush(b1PushCallback);
  b2.attachPush(b2PushCallback);
  b3.attachPush(b3PushCallback);
  b4.attachPush(b4PushCallback);
  b5.attachPush(b5PushCallback);
  b6.attachPush(b6PushCallback);
  b7.attachPush(b7PushCallback);
  b8.attachPush(b8PushCallback);

  b10.attachPush(b10PushCallback);  //nr 0
  b11.attachPush(b11PushCallback);  //#
  b12.attachPush(b12PushCallback);  //CLEAR
  b13.attachPush(b13PushCallback);  //PRE-test
  //b14.attachPush(b14PushCallback);  //PRE-test
  //b15.attachPush(b15PushCallback);  //PRE-test
 // b16.attachPush(b16PushCallback);  //PRE-test

  pinMode(13, OUTPUT);  // define,  pin 13 is a builtin green LED on uno

  smartDelay(1000);

}

void loop() {  // Put your main code here, to run repeatedly:
   //if (Serial.available())     //for debug, look at the serialmonitor from arduino
   //Serial.write(GSMSerial.read());

   nexLoop(nex_events_call);  // Check for any touch event
/*
//check for incoming sms
if (gsmSms.messageAvailable())
    {
        char receivedNumber[30] = "";
        gsmSms.receivedNumber(receivedNumber, 30);  // Phone number of the sender.
        Serial.println(F("Message received from: "));
        Serial.println(receivedNumber);
        char receivedContent[165] = "";
        gsmSms.receivedContent(receivedContent, 165);   // Contents of the SMS.
        Serial.println(F("Message contains: "));
        Serial.println(receivedContent);
        gsmSms.messageFlush();  // Delete 'read', 'sent' and 'saved but unsent' messages.
    }
*/
    PushNex(str);
    PushNexP2(Number);
    PushNexP2(SMSTxt);
    PushNexP3(str3);   //quotes data
    Serial.println(F(""));
/*
//GSP info running text
Serial.print("g0.txt=");  // This is sent to the nextion display to set what object name (before the dot) and what
atribute (after the dot) are you going to change.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.print(str);  // This is the text you want to send to that object and atribute mentioned before.
    Serial.print("\"");  // Since we are sending text, and not a number, we need to send double quote before and after
the actual text.
    Serial.write(0xff);  // We always have to send this three lines after each command sent to the nextion display.
    Serial.write(0xff);
    Serial.write(0xff);
    smartDelay(1000);

*/
 smartDelay(1000);
}
```

```
void readln()
{
  str3 = gprs.readString();
  Serial.println(str3);

}

static void smartDelay(unsigned long ms)
{
  unsigned long start = millis();
  while (millis() - start < ms);
}
```

mobile.HMI

Bạn code dùng cái editor (xem HMI bên trên), sau khi xong bạn save cái output (nó là binary khong phải ASCII) vào SD card. Cho SD card vào HMI display (bên cạnh có cái slot), power off, power on. DONE.

Cái HMI của tôi



Name	Ext	Size	Date
[..]		<DIR>	10/02/2019 21.30
[gsmphone]		<DIR>	23/02/2019 14.34
mobile	HMI	2.593.468	06/04/2019 19.58

Hình của phone display HMI


b17
BACK
t1
Tphone
b0
1
b1
2
b2
3
b3
4
b4
5
b5
6
b6
7
b7
8
b8
9
b9
*
b10
0
b11
#
b12
CLEAR
b13
On my way
b14
Cant talk
b15
Out to lunch
b16
No problem
m0
m1
SMS
0 page0
1 page1
2 page2
属性
b0[按钮]
type 59
id 2
objname b0
vscope 私有
sta 单色
style 3D_Auto
font 0
bco 48631
事件
按下事件(1)
弹起事件(0)
发送键值


b0
BACK
t3
Read internet from GPRS via GSM
0 page0
1 page1
2 page2
属性
page2[页面]
type 65