**CODE TO CONNECT TO THE SERVER**

#include <SoftwareSerial.h>

// SoftwareSerial Serial(7, 8); // RX, TX

enum \_parseState {

PS\_DETECT\_MSG\_TYPE,

PS\_IGNORING\_COMMAND\_ECHO,

PS\_HTTPACTION\_TYPE,

PS\_HTTPACTION\_RESULT,

PS\_HTTPACTION\_LENGTH,

PS\_HTTPREAD\_LENGTH,

PS\_HTTPREAD\_CONTENT

};

byte parseState = PS\_DETECT\_MSG\_TYPE;

char buffer[80];

byte pos = 0;

int contentLength = 0;

void resetBuffer() {

memset(buffer, 0, sizeof(buffer));

pos = 0;

}

void sendSerial(const char\* msg, int waitMs = 500) {

Serial.println(msg);

delay(waitMs);

while (Serial.available()) {

parseATText(Serial.read());

}

}

void setup()

{

// Serial.begin(9600);

Serial.begin(9600);

sendSerial("AT+SAPBR=3,1,\"APN\",\"airtelgprs.com\"");

sendSerial("AT+SAPBR=1,1", 3000);

sendSerial("AT+HTTPINIT");

sendSerial("AT+HTTPPARA=\"CID\",1");

sendSerial("AT+HTTPPARA=\"URL\",\"http://www.iforce2d.net/test.php\"");

sendSerial("AT+HTTPACTION=0");

}

void loop()

{

while (Serial.available()) {

// Serial.println("inside while loop-loop");

parseATText(Serial.read());

}

}

void parseATText(byte b) {

buffer[pos++] = b;

if ( pos >= sizeof(buffer) )

resetBuffer(); // just to be safe

/\*

// Detailed debugging

Serial.println();

Serial.print("state = ");

Serial.println(state);

Serial.print("b = ");

Serial.println(b);

Serial.print("pos = ");

Serial.println(pos);

Serial.print("buffer = ");

Serial.println(buffer);\*/

switch (parseState) {

Serial.println ("inside switch case");

case PS\_DETECT\_MSG\_TYPE:

{

if ( b == '\n' )

resetBuffer();

else {

if ( pos == 3 && strcmp(buffer, "AT+") == 0 ) {

parseState = PS\_IGNORING\_COMMAND\_ECHO;

}

else if ( b == ':' ) {

//Serial.print("Checking message type: ");

//Serial.println(buffer);

if ( strcmp(buffer, "+HTTPACTION:") == 0 ) {

Serial.println("Received HTTPACTION");

parseState = PS\_HTTPACTION\_TYPE;

}

else if ( strcmp(buffer, "+HTTPREAD:") == 0 ) {

Serial.println("Received HTTPREAD");

parseState = PS\_HTTPREAD\_LENGTH;

}

resetBuffer();

}

}

}

break;

case PS\_IGNORING\_COMMAND\_ECHO:

{

if ( b == '\n' ) {

Serial.print("Ignoring echo: ");

Serial.println(buffer);

parseState = PS\_DETECT\_MSG\_TYPE;

resetBuffer();

}

}

break;

case PS\_HTTPACTION\_TYPE:

{

if ( b == ',' ) {

Serial.print("HTTPACTION type is ");

Serial.println(buffer);

parseState = PS\_HTTPACTION\_RESULT;

resetBuffer();

}

}

break;

case PS\_HTTPACTION\_RESULT:

{

if ( b == ',' ) {

Serial.print("HTTPACTION result is ");

Serial.println(buffer);

parseState = PS\_HTTPACTION\_LENGTH;

resetBuffer();

}

}

break;

case PS\_HTTPACTION\_LENGTH:

{

if ( b == '\n' ) {

Serial.print("HTTPACTION length is ");

Serial.println(buffer);

// now request content

Serial.print("AT+HTTPREAD=0,");

Serial.println(buffer);

parseState = PS\_DETECT\_MSG\_TYPE;

resetBuffer();

}

}

break;

case PS\_HTTPREAD\_LENGTH:

{

if ( b == '\n' ) {

contentLength = atoi(buffer);

Serial.print("HTTPREAD length is ");

Serial.println(contentLength);

Serial.print("HTTPREAD content: ");

parseState = PS\_HTTPREAD\_CONTENT;

resetBuffer();

}

}

break;

case PS\_HTTPREAD\_CONTENT:

{

// for this demo I'm just showing the content bytes in the serial monitor

Serial.write(b);

contentLength--;

if ( contentLength <= 0 ) {

// all content bytes have now been read

parseState = PS\_DETECT\_MSG\_TYPE;

resetBuffer();

}

}

break;

}

}