

Módulo 12

GSM / GPRS

Tecnologías Emergentes

Índice

- El contenido de esta unidad es el siguiente:
 - Introducción
 - Objetivos
 - Temas a desarrollar:
 - Lección 1: GSM
 - Lección 2: GPRS
 - Bibliografía

Introducción

- En esta unidad se estudiará los conceptos asociados a la comunicación a través de celular.

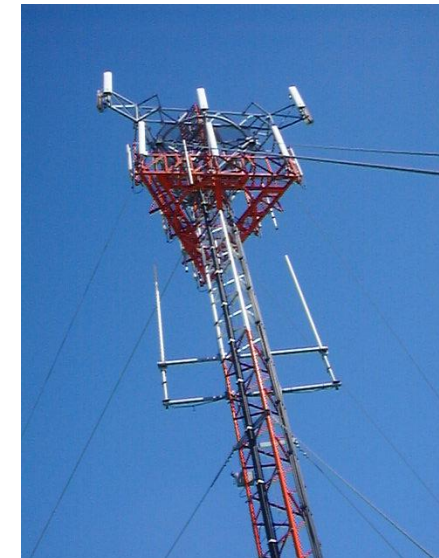
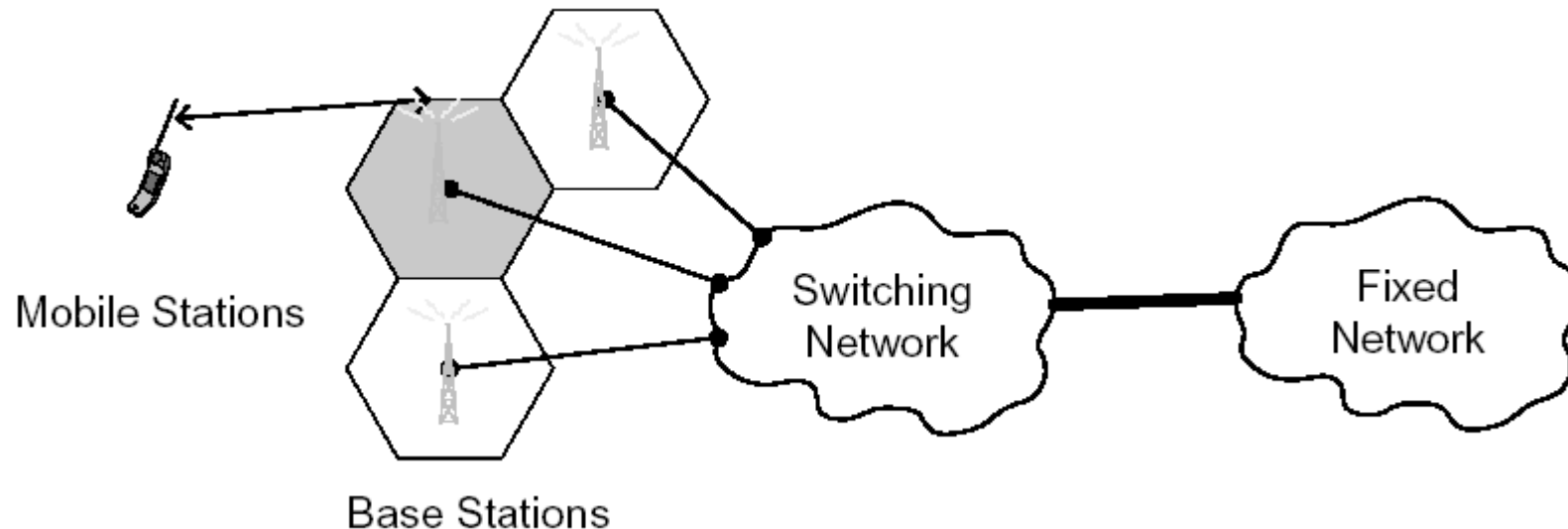
Objetivos

- Al finalizar la unidad el estudiante será capaz de:
 - Explicar los conceptos básicos utilizados en la comunicación a través de una red celular.

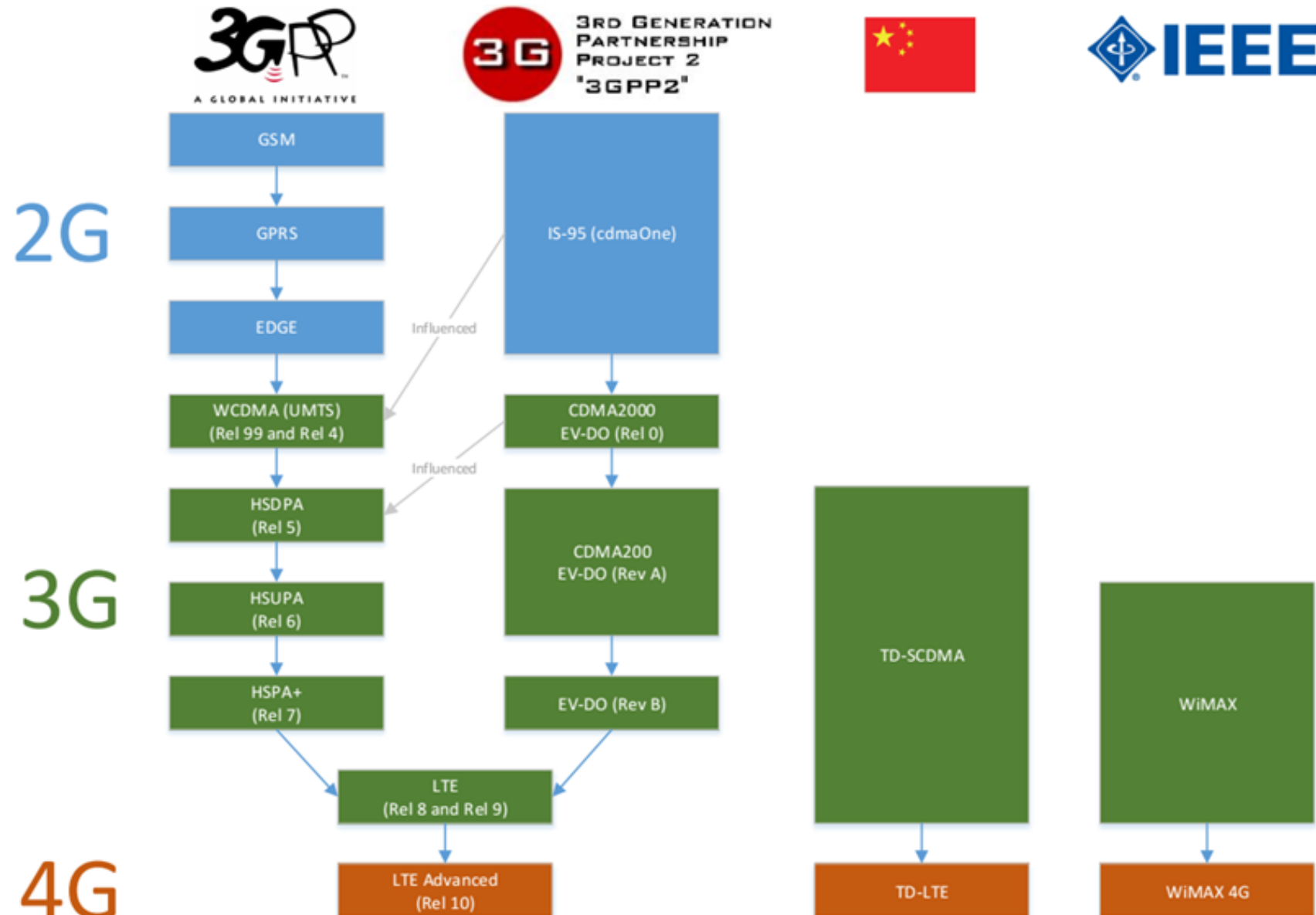
Tecnologías Celular

Telefonía móvil

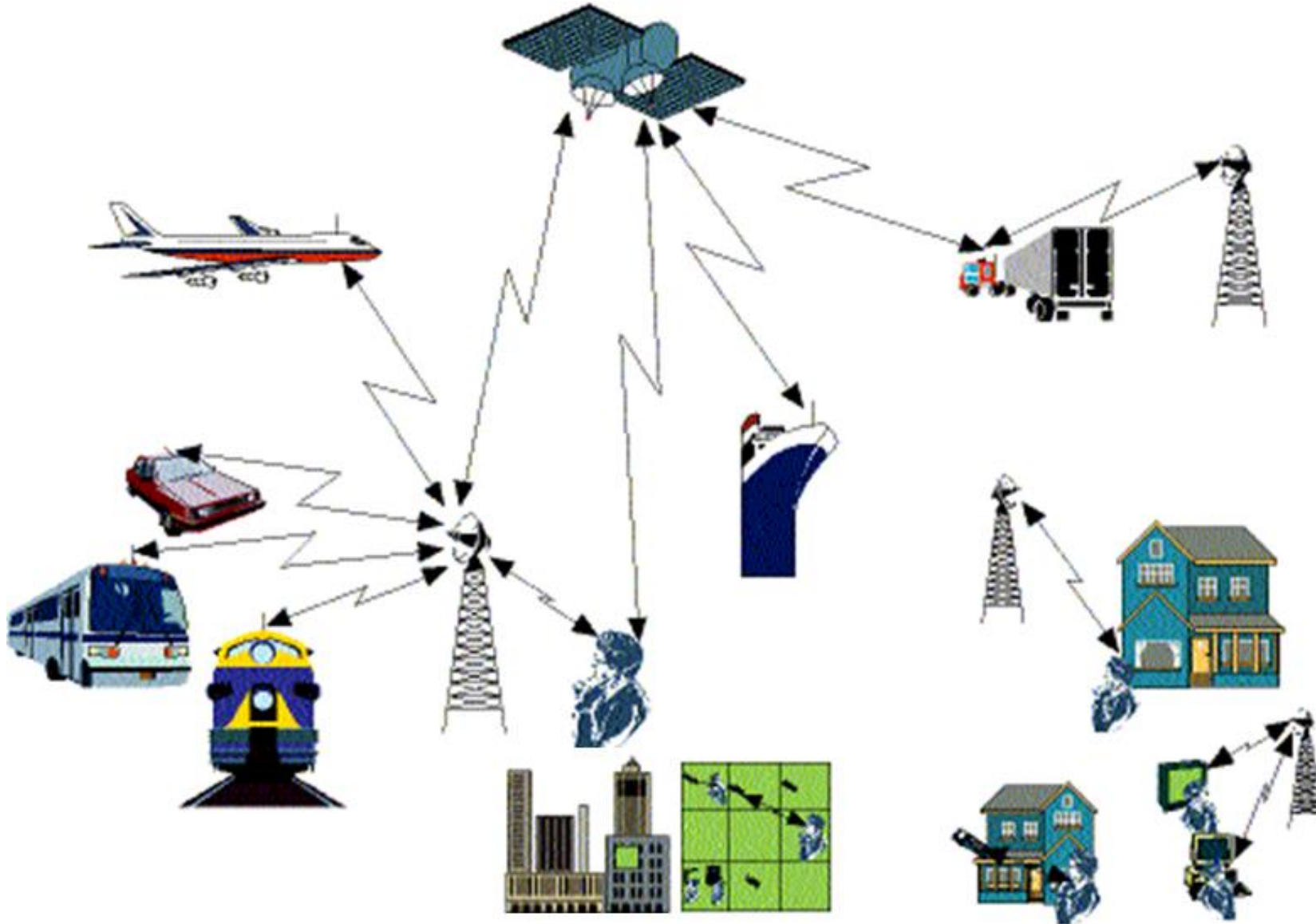
- La telefonía móvil o telefonía celular es un medio de comunicación inalámbrico a través de ondas electromagnéticas.
- Como cliente se utiliza un dispositivo denominado teléfono móvil o teléfono celular.



Tecnologías móviles



Uso de la telefonía móvil



GSM

- El sistema global para las comunicaciones móviles (del inglés Global System for Mobile communications, GSM, y originariamente del francés Groupe Spécial Mobile) es un sistema estándar, libre de regalías, de telefonía móvil digital.
- Por su velocidad y características se considera de 2da generación (2G).
- Es la más utilizada en el mundo.

Servicios de GSM

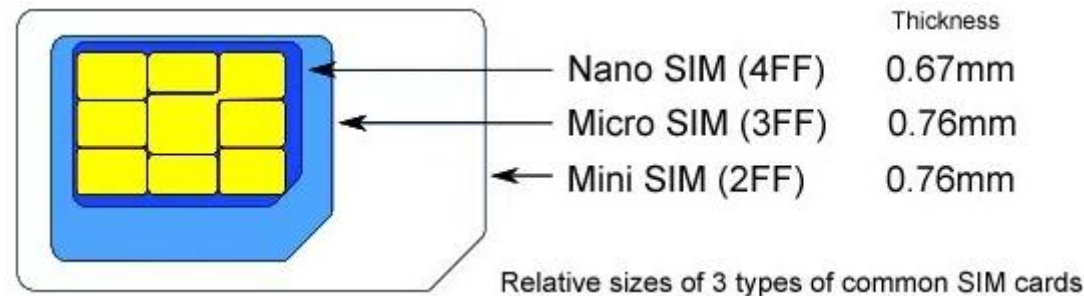
- GSM ofrece los siguientes servicios:
 - Teleservicios: servicios básicos de telefonía (Telefonía, llamadas de Emergencia, SMS, servicios de fax y voz).
 - Servicios Portadores: Transmisión / Recepción síncrona y asíncrona de datos.
 - Servicios Complementarios: llamada en espera, llamadas múltiples, identificación de llamada.

Frecuencias usadas por GSM

Banda0	Nombre	Notas
GSM 850	GSM 850	Usada en los EE.UU., Sudamérica y Asia.
GSM 900	P-GSM 900	La banda con que nació GSM en Europa y la más extendida
	E-GSM 900	E-GSM, extensión de GSM 900
	R-GSM 900	GSM ferroviario (GSM-R)
GSM1800	GSM 1800	
GSM1900	GSM 1900	Usada en Norteamérica, incompatible con GSM-1800 por solapamiento de bandas

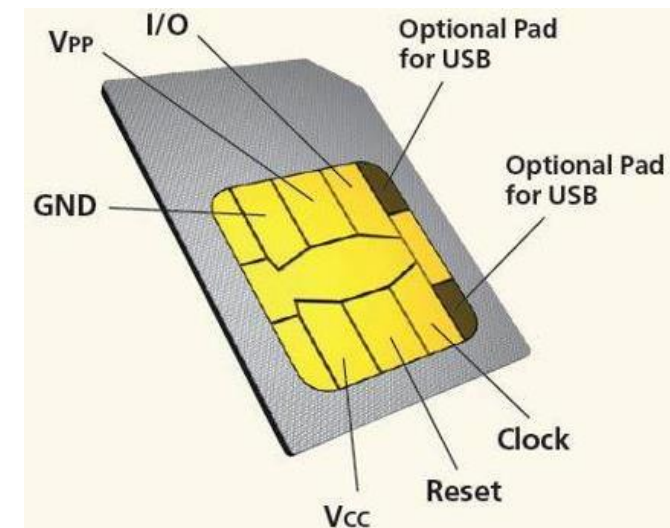
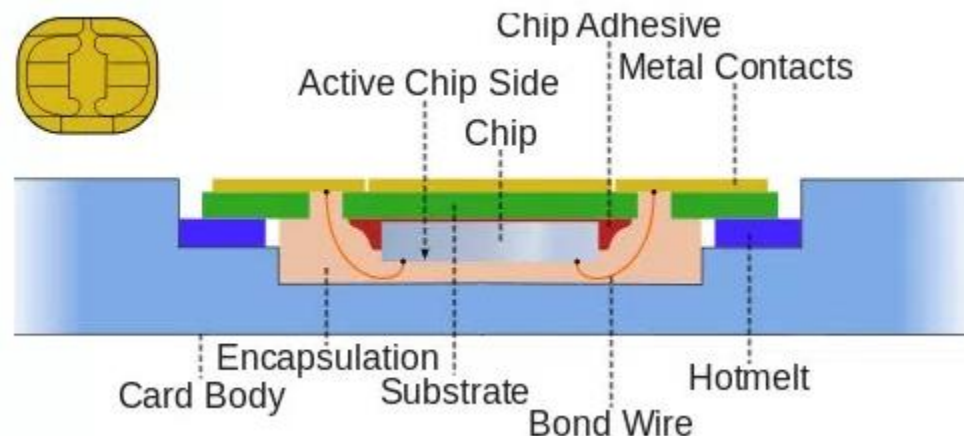
Tarjeta SIM

- Una tarjeta SIM (acrónimo en inglés de Subscriber Identity Module, en español módulo de identificación de suscripción) es una tarjeta inteligente desmontable usada en teléfonos móviles y módems HSPA o LTE que se conectan al puerto USB.



Tarjeta SIM

- Las tarjetas SIM almacenan de forma segura la clave de servicio del suscriptor usada para identificarse ante la red, de forma que sea posible cambiar la suscripción del cliente de un terminal a otro simplemente cambiando la tarjeta.



Tarjeta SIM

- El uso de la tarjeta SIM es obligatorio en las redes GSM.
- En las redes UMTS se denomina USIM o UICC (acrónimo de Universal Integrated Circuit Card, Tarjeta Universal de Circuito Integrado)
- En las redes CDMA se denomina RUIM (Removable User Identify Module, 'Módulo de Identidad de Usuario Desmontable').



GPRS

GPRS

- El servicio general de paquetes vía radio, en inglés: General Packet Radio Service (GPRS), fue creado en la década de los 80.
- Es una extensión mejorada del GSM.
 - GSM: 5-10 mensajes por minuto, GPRS: 30 mensajes por minuto.
- Con GPRS se pueden utilizar servicios como Wireless Application Protocol (WAP), servicio de mensajes cortos (SMS), Multimedia Messaging System (MMS), Internet y para los servicios de comunicación, como el correo electrónico y la World Wide Web (WWW).

Conexión GPRS

- Una conexión GPRS está establecida por la referencia a su nombre de punto de acceso (APN).

Nombre : Claro Datos

APN: **claro.pe**

Nombre de usuario: claro

Contraseña: claro

Tipo de autenticación: PAP

Tipo de apn: default

Nombre: Movistar Internet

APN: **movistar.pe**

Nombre de usuario: movistar@datos

Contraseña: movistar

MCC: 716

MNC: 06

Tipo de APN: Internet

Comparación

Symbol	Standard	Full Name	Maximum Download Speed (Theoretical)	Maximum Upload Speed (Theoretical)
2G	GSM	Global System for Mobile Communications	14.4 Kbits/s	14.4 Kbits/s
G	GPRS	General Packet Radio Service	53.6 Kbits/s	26.8 Kbits/s
E	EDGE	Enhanced Data rates for GSM Evolution	217.6 Kbits/s	108.8 Kbits/s
3G	UMTS	Universal Mobile Telecommunications System	384 Kbits/s	128 Kbits/s
H	HSPA	High-Speed Packet Access	7.2 Mbits/s	3.6 Mbits/s
H+	HSPA+	Evolved High-Speed Packet Access - Release 6	14.4 Mbits/s	5.76 Mbits/s
H+	HSPA+	Evolved High-Speed Packet Access - Release 7	21.1 Mbits/s or 28.0 Mbits/s	11.5 Mbits/s
H+	HSPA+	Evolved High-Speed Packet Access - Release 8	42.2 Mbits/s	11.5 Mbits/s
H+	HSPA+	Evolved High-Speed Packet Access - Release 9	84.4 Mbits/s	11.5 Mbits/s
H+	HSPA+	Evolved High-Speed Packet Access - Release 10	168.8 Mbits/s	23.0 Mbits/s
4G	LTE	Long Term Evolution	100 Mbits/s	50 Mbits/s
4G	LTE-A	Long Term Evolution - Advanced	1 Gbits/s	500 Mbits/s

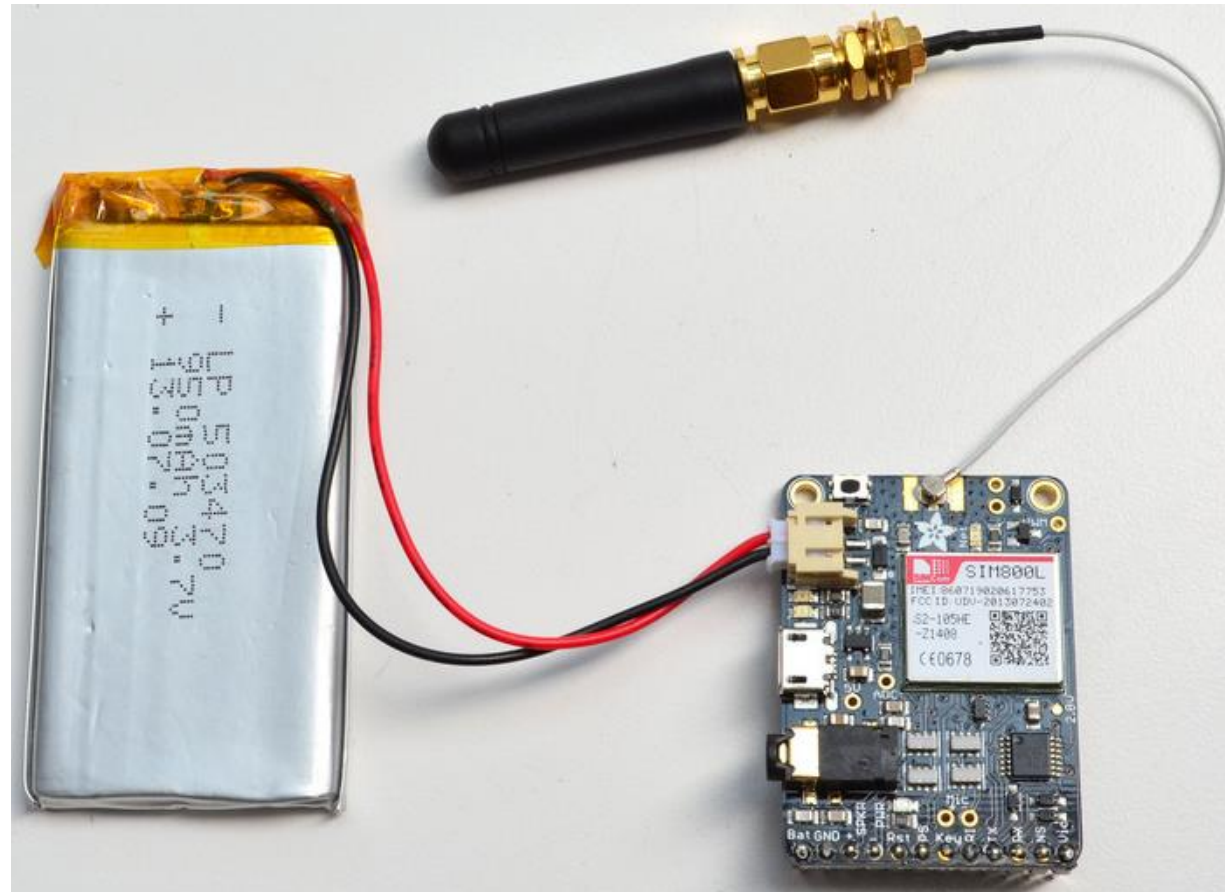
GSM/GPRS para Arduino

GSM Shield

- Existen diferentes opciones:



Conexión



Conexión



Código para Conexión

```
1 //Programa para comunicación con Modem vía puerto serial
2 //por software
3 #include <SoftwareSerial.h>
4
5 long velocidad=9600;
6 SoftwareSerial Modem(2, 3); // Pin 2 es RX | Pin 3 es TX
7
8 void setup()
9 {
10     Serial.begin(velocidad);
11     Modem.begin(velocidad);
12     Serial.println("Listo para comandos AT ..");
13 }
14
15 void loop()
16 {
17     if (Modem.available())
18     { char c = Modem.read() ;
19       Serial.print(c);
20     }
21     if (Serial.available())
22     { char c = Serial.read();
23       Modem.print(c);
24     }
25 }
```

Comandos AT

ATI

Quectel_Ltd
Quectel_M10
Revision: M10FAR01A02

OK

AT+CFUN?

+CFUN: 1

OK

AT+CPIN?

+CPIN: READY

OK

AT+COPS?

+COPS: 0,0,"CLARO PER"

OK

*AT+CFUN=1
AT+CPIN="1234"
AT+COPS?*

AT - SIM

AT+GSN

868326020259115

OK

AT+CIMI

716101601681907

OK

AT+CPOL?

+CPOL: 1,2,"71610",0,0,1

+CPOL: 2,2,"71610",1,0,0

+CPOL: 3,2,"722310",0,0,1

+CPOL: 4,2,"722310",1,0,0

+CPOL: 5,2,"72405",0,0,1

AT+QBAND?

+QBAND: "GSM850_EGSM_DCS_PCS_MODE"

OK

EGSM_MODE"

"DCS_MODE"

"PCS_MODE"

"GSM850_MODE"

"EGSM_DCS_MODE"

"GSM850_PCS_MODE"

"GSM850_EGSM_DCS_PCS_MODE"

AT - PIN

AT+CPIN?

+CPIN: SIM PIN
OK

//Query PIN code is locked

AT+CPIN=1234

//PIN

+CPIN: READY
OK

AT+CPIN?

//PIN has already been entered

+CPIN: READY
OK

AT+CPIN?

+CPIN: SIM PUK
OK

//Query PUK code is locked

AT+CPIN="26601934","1234"

//Enter PUK and new PIN password

+CPIN: READY
OK

AT+CPIN?

+CPIN: READY
OK

//PUK has already been entered

AT – Llamadas de voz

```
ATD+51999407034;
```

```
OK
```

```
ATH
```

```
OK
```

```
RING
```

```
RING
```

```
AT+CLCC
```

```
+CLCC: 1,1,4,0,0,"013173900",129,""
```

```
OK
```

```
RING
```

```
RING
```

```
ATA
```

```
OK
```

```
ATH
```

```
OK
```

AT – Mensajes SMS

```
AT+CMGF=1
```

```
OK
```

```
AT+CMGL="ALL"
```

```
+CMGL: 1,"REC UNREAD","+131","", "2017/06/03 18:39:23-20"
```

```
Tu casilla de voz esta casi llena, elimina los mensajes que ya escuchaste pa
```

```
+CMGL: 2,"REC UNREAD","+131","", "2017/06/04 09:49:11-20"
```

```
Tu casilla de voz esta casi llena, elimina los mensajes que ya escuchaste pa
```

```
+CMGL: 3,"REC UNREAD","CLARO","", "2017/11/07 09:50:30-20"
```

```
NO ESPERES! Por S/10 mas a tu plan, obten 1 GB adicionales de Internet y 200
```

```
+CMGL: 5,"REC READ","", "", "2017/11/09 10:14:42-20"
```

```
Msje Claro:
```

```
Te llame el dia 09/11 a las 09:46 (1 llamadas)
```

```
OK
```

AT – Mensajes SMS

[illegible]

```
AT+CMGR=3 //Read message
+CMGR: "REC UNREAD","+8615021012496","", "2010/09/25
15:06:37+32",145,4,0,241,"+8
613800210500",145,27
This is a test from Quectel
OK
```

```
AT+CMGF=1 //Set SMS message format as text mode
OK
```

```
AT+CSCS="GSM" //Set character set as GSM
OK
```

AT+CMGS="15021012496"

> This is a test from Quectel

CTRL+Z o ESC

+CMGS: 247

OK

Sniffer Web

- Comunicación entre el cliente y el servidor web

```
GET /webacs.htm HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.5 [es] (Win95; I)
Host: www.tecsup.edu.pe
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: es
Accept-Charset: iso-8859-1,*,utf-8
```

```
HTTP/1.0 200 OK
Date: Mon, 27 Sep 1999 21:20:16 GMT
Server: Novell-HTTP-Server/3.1R1
Content-type: text/html
Last-modified: Mon, 06 Jul 1998 06:12:56 GMT
Content-length: 923
```

```
<HTML>
<HEAD>
<TITLE>GroupWise WebAccess</TITLE>
</HEAD>
<BODY bgcolor="#FFFFFF" link="#000000" vlink="#000000" alink="#000000">
<CENTER>
<TABLE cellspacing=0 cellpadding=0 width=min border=0>
<TR>
<TD valign=top width=min align=left></TD>
</TR></TABLE>
</CENTER>
</BODY></HTML>
```

Servidor Web en Arduino



```
GET / HTTP/1.1
Host: 192.168.115.4
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.73 Safari
DNT: 1
Accept-Encoding: gzip, deflate
Accept-Language: en-US
```

Cada línea termina en `\r\n`

La última línea está en blanco y termina en `\r\n`

Servidor Web

```
1 // Programa para crear un servidor Web
2 #include <GSM.h>
3
4 // PIN Number
5 #define PINNUMBER "1234"
6
7 // APN data
8 #define GPRS_APN      "claro.pe" // replace your GPRS APN
9 #define GPRS_LOGIN    "claro"    // replace with your GPRS Login
10 #define GPRS_PASSWORD "claro" // replace with your GPRS password
11
12 // initialize the library instance
13 GPRS gprs;
14 GSM gsmAccess;
15 GSMServer server(80); // port 80 (http default)
16
17 // timeout
18 const unsigned long __TIMEOUT__ = 10 * 1000;
19
```

```
20 void setup() {
21     // initialize serial communications
22     Serial.begin(9600);
23     while (!Serial) {
24         ; // wait for serial port to connect.
25     }
26
27     // connection state
28     boolean notConnected = true;
29
30     // Start GSM shield
31     // If your SIM has PIN, pass it as a parameter of begin() in quotes
32     while (notConnected) {
33         if ((gsmAccess.begin(PINNUMBER) == GSM_READY) &
34             (gprs.attachGPRS(GPRS_APN, GPRS_LOGIN, GPRS_PASSWORD) == GPRS_READY)) {
35             notConnected = false;
36         } else {
37             Serial.println("Not connected");
38             delay(1000);
39         }
40     }
41
42     Serial.println("Connected to GPRS network");
43
44     // start server
45     server.begin();
46
47     //Get IP.
48     IPAddress LocalIP = gprs.getIPAddress();
49     Serial.println("Server IP address=");
50     Serial.println(LocalIP);
51 }
52
```

```
53 void loop() {
54     // listen for incoming clients
55     GSMClient client = server.available();
56
57     if (client) {
58         while (client.connected()) {
59             if (client.available()) {
60                 Serial.println("Receiving request!");
61                 bool sendResponse = false;
62                 while (char c = client.read()) {
63                     if (c == '\n') {
64                         sendResponse = true;
65                     }
66                 }
67                 if (sendResponse) {
68                     // send a standard http response header
69                     client.println("HTTP/1.1 200 OK");
70                     client.println("Content-Type: text/html");
71                     client.println();
72                     client.println("<html>");
73                     // output the value of each analog input pin
74                     for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
75                         client.print("analog input ");
76                         client.print(analogChannel);
77                         client.print(" is ");
78                         client.print(analogRead(analogChannel));
79                         client.println("<br />");
80                     }
81                     client.println("</html>");
82                     delay(1000);
83                     client.stop();
84                 }
85             }
86         }
87     }
88 }
```

Enviar mensaje SMS

```
1  // Programa para enviar mensaje SMS
2  #include <GSM.h>
3  #define PINNUMBER "1234"
4
5  GSM gsmAccess;
6  GSM_SMS sms;
7
8  void setup() {
9      // initialize serial communications and wait for port to open
10     Serial.begin(9600);
11     while (!Serial) {
12         ; // wait for serial port to connect
13     }
14
15     Serial.println("SMS Messages Sender");
16
17     // connection state
18     boolean notConnected = true;
19
20     // Start GSM shield
21     while (notConnected) {
22         if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
23             notConnected = false;
24         } else {
25             Serial.println("Not connected");
26             delay(1000);
27         }
28     }
29     Serial.println("GSM initialized");
30 }
31
```

```
32 void loop() {
33     Serial.print("Enter a mobile number: ");
34     char remoteNum[20]; // telephone number to send sms
35     readSerial(remoteNum);
36     Serial.println(remoteNum);
37
38     // sms text
39     Serial.print("Now, enter SMS content: ");
40     char txtMsg[200];
41     readSerial(txtMsg);
42     Serial.println("SENDING"); Serial.println();
43     Serial.println("Message:");
44     Serial.println(txtMsg);
45
46     // send the message
47     sms.beginSMS(remoteNum);
48     sms.print(txtMsg);
49     sms.endSMS();
50     Serial.println("\nCOMPLETE!\n");
51 }
52
53 // Read input serial
54 int readSerial(char result[]) {
55     int i = 0;
56     while (1) {
57         while (Serial.available() > 0) {
58             char inChar = Serial.read();
59             if (inChar == '\n') {
60                 result[i] = '\0';
61                 Serial.flush();
62                 return 0;
63             }
64             if (inChar != '\r') {
65                 result[i] = inChar;
66                 i++;
67             }
68         }
69     }
70 }
```

Recibir mensaje SMS

```
1 // Programa para Leer mensaje SMS
2 #include <GSM.h>
3 #define PINNUMBER "1234"
4
5 GSM gsmAccess;
6 GSM_SMS sms;
7
8 // Array to hold the number a SMS is retrieved from
9 char senderNumber[20];
10
```

```
11 void setup() {
12     // initialize serial communications and wait for port to open
13     Serial.begin(9600);
14     while (!Serial) {
15         ; // wait for serial port to connect
16     }
17
18     Serial.println("SMS Messages Receiver");
19
20     // connection state
21     boolean notConnected = true;
22
23     // Start GSM connection
24     while (notConnected) {
25         if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
26             notConnected = false;
27         } else {
28             Serial.println("Not connected");
29             delay(1000);
30         }
31     }
32
33     Serial.println("GSM initialized");
34     Serial.println("Waiting for messages");
35 }
36
```



```
37 void loop() {
38     char c;
39
40     // If there are any SMSs available()
41     if (sms.available()) {
42         Serial.println("Message received from:");
43
44         // Get remote number
45         sms.remoteNumber(senderNumber, 20);
46         Serial.println(senderNumber);
47
48         // An example of message disposal
49         // Any messages starting with # should be discarded
50         if (sms.peek() == '#') {
51             Serial.println("Discarded SMS");
52             sms.flush();
53         }
54
55         // Read message bytes and print them
56         while (c = sms.read()) {
57             Serial.print(c);
58         }
59
60         Serial.println("\nEND OF MESSAGE");
61
62         // Delete message from modem memory
63         sms.flush();
64         Serial.println("MESSAGE DELETED");
65     }
66
67     delay(1000);
68 }
```

Arduino y 4G



4G + GPS Shield for Arduino
Cooking Hacks

- Output power
 - Class 4 (2 W, 33 dBm) @ GSM 900
 - Class 1 (1 W, 30 dBm) @ GSM 1800
 - Class E2 (0.5 W, 27 dBm) @ EDGE 900
 - Class E2 (0.4 W, 26 dBm) @ EDGE 1800
 - Class 3 (0.25 W, 24 dBm) @ UMTS
 - Class 3 (0.2 W, 23 dBm) @ LTE
- Data Transmission:
 - LTE:
 - Downlink up to 100 Mbps
 - Uplink up to 50 Mbps
 - HSPA+:
 - Downlink up to 42.0Mbps
 - Uplink up to 5.76Mbps
 - UMTS:
 - Uplink/Downlink up to 384 kbps
- Protocols:
 - TCP/UDP
 - HTTP
 - FTP

Resumen

- La placa Arduino puede comunicarse con otros dispositivos a través de Ethernet, Bluetooth, Wifi y, ahora, GSM.
- El uso de GSM permite una comunicación a nivel mundial, pero requiere de una tarjeta SIM.

Bibliografía

- Programming Arduino Getting Started with Sketches 1st Edition. Simon Monk 2012.
- Exploring Arduino: Tools and Techniques for Engineering Wizardry 1st Edition, Jeremy Blum 2013.
- Arduino Workshop: A Hands-On Introduction with 65 Projects 1st Edition John Boxall 2013.

Fin de la unidad