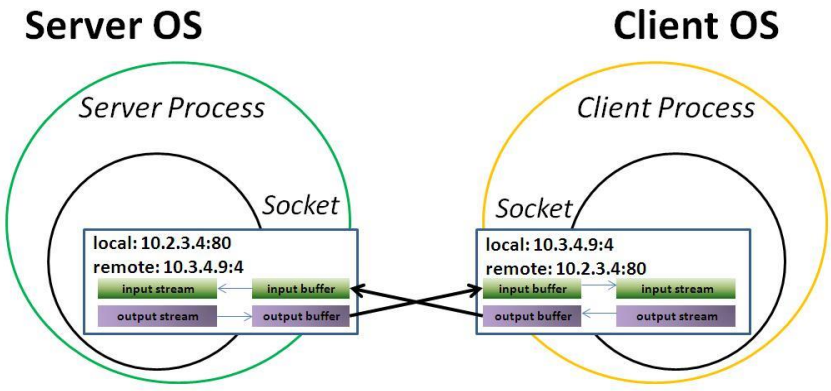
## Comunicación por TCP usando sockets.

Prof. Arturo Ocampo Álvarez
FES Aragón, UNAM

Un Socket (enchufe) generalmente se basa en la arquitectura cliente/servidor. Para las comunicaciones TCP (Protocolo de control de transmisión), un host (anfitrion) escucha las solicitudes de conexión entrantes. Cuando llega una solicitud, el host del servidor la acepta, en cuyo momento los datos pueden transferirse entre los hosts. UDP (Protocolo de datagramas de usuario) también puede establecer una conexión, aunque no es obligatorio. Los datos pueden simplemente enviarse o recibirse desde un host.

Todas las pilas TCP/IP tienen 65,536 puertos para TCP y UDP. Hay un complemento completo de puertos para UDP (numerado 0-65535) y otro complemento completo, con el mismo esquema de numeración, para TCP. ***Un puerto no es una interfaz física***, es un concepto que simplifica las comunicaciones de Internet para los seres humanos. Al recibir un paquete, la pila de protocolos lo dirige al puerto específico. Si no hay ninguna aplicación escuchando en ese puerto, el paquete se descarta y se puede devolver un error al remitente. Sin embargo, las aplicaciones pueden crear sockets, que les permiten conectarse a un puerto. Una vez que una aplicación haya creado un socket y lo haya vinculado a un puerto, los datos destinados a ese puerto se enviarán a la aplicación. Por eso se usa el término socket: es el mecanismo de conexión entre el mundo exterior (los puertos) y la aplicación.




## Actividad 5.4.

a) Abra una ventana de terminal MSYS2 MINGW32 ejecutando:

C:\msys32\mingw32.exe .

b) Copiar el directorio “tcp_server” y "scripts"

$ export IDF_PATH="C:/msys32/home/USUARIO/esp/esp-idf"

$ cd esp

$ cp -r $IDF_PATH/examples/protocols/sockets/tcp_server .

$ cp -r $IDF_PATH/examples/protocols/sockets/scripts .

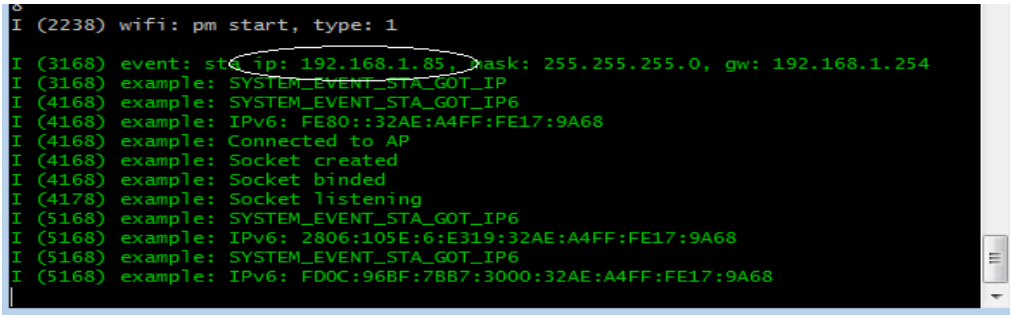
$ cd tcp_server

c) Configurar el puerto y en "Example Configuration" asignar el SSID y el password de la red.

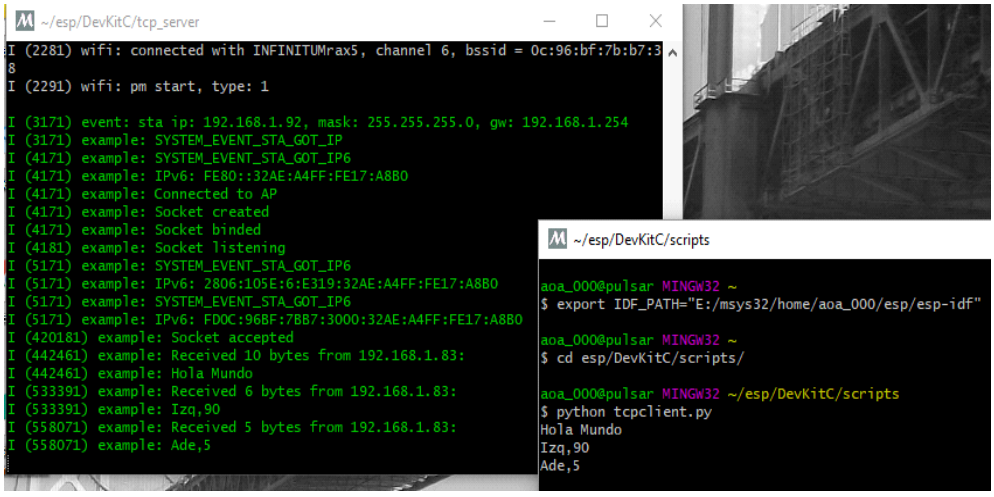
$ make menuconfig

$ make flash monitor

d) Localizar el IP asignado al servidor




e) Desde otra ventana mingw32 u otra computadora Editar el archivo "tcpclient.py" y asignar el IP del servidor. Mandar un mensaje!




**Experimento 4.**

El grupo de investigación REEX de la FES Aragón, requiere accionar un grupo de Robots de enjambre usando la comunicación TCP/IP para realizar una acción de control a distancia.

Construcción del modelo.

Un equipo de estudiantes desarrolla la segunda etapa:

V. Modificar el programa "tcp_server.c" para que al recibir un mensaje del cliente se ejecute una acción (encender un Led, activar un Motor, etc.).

```
/* ----Agregar despues del if (linea 192)--------------------*/
                else {
                 char *token = strtok(rx_buffer,",");
                 char delimitador[] = ",";
                 char instruccion[10], dato[10];
                 strcpy(instruccion,token);
                 token = strtok(NULL, delimitador);
                 strcpy(dato,token);

                 ESP_LOGI(TAG, "%s", dato);

                }
/* -------------------------------------------------------*/
```

VI. Una vez que ya se tiene la instrucción y el dato por separado, se puede utilizar un “if” o “switch”, para determinar la instrucción y el numero de veces que se debe repetir.

Documentar el modelo y proceso.

¿Cuáles son las herramientas que selecciono y por qué?

¿Qué herramienta matemática utilizo?

¿Qué características se deben considerar para instalar el entorno de programación más adecuado?

¿Qué fuentes y recursos utilizo?

Refinación mediante autoevaluación

Notifique las problemáticas encontradas y su solución.

Especifique las consideraciones de seguridad, costo y técnicas requeridas.

Generación del Modelo.

Se presentan resultados y posibles mejoras.

Efectividad

Análisis de costos y portabilidad.

Conclusiones técnicas, éticas y oportunidades.