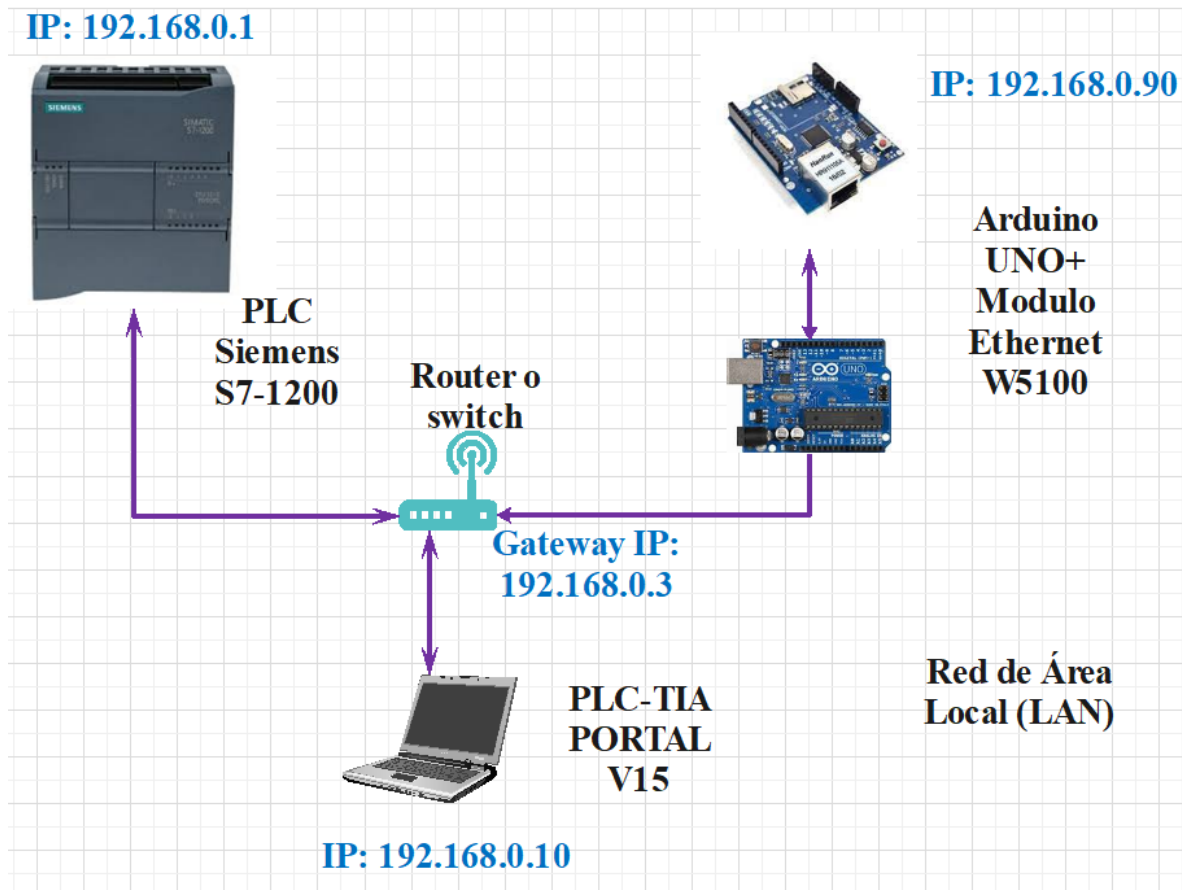


# 1. Esquema de red para usar protocolo Siemens s7 con *Settimino*



## 2. S7-Ethernet protocol (Siemens S7-Protocol)

1. El protocolo S7 se basa en el estándar ISP TCP (RFC1006), es decir, los datos se transfieren mediante “bloques”.
2. Cada bloque se denomina PDU (Protocol Data Unit) con un tamaño fijo entre 240 bytes y 960 bytes.
3. El protocolo S7 es **orientado en función** o **orientado en comando**. Cada transmisión contiene o un comando o una replica de este comando en caso de renegociación por fallo en transmisión, PDU con tamaño incoherente (superior o inferior a los bytes indicados), etc.

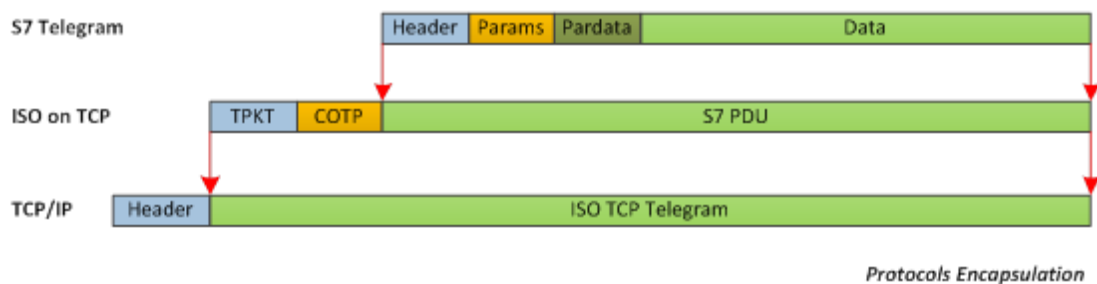


Fig.1. Esquema de protocolo Siemens S7. Tomado de <http://settimino.sourceforge.net/>

4. Cada comando (PDU) consiste en los siguientes elementos: **Un encabezado (Header)+Un conjunto de parámetros+ los datos de los parámetros+ bloque de datos**. Los comandos están divididos en las siguientes categorías:

- Escritura/Lectura de datos
- Escritura/Lectura de datos cíclicos
- Información de directorio
- Información de sistema (Por ejemplo, PLC Status)
- Movimiento de bloques
- Control de PLC
- Fecha y hora
- Seguridad
- Programación

### 3. Usar Protocolo Siemens S7 con *Settimino*

- Descargar la librería (<https://sourceforge.net/projects/settimino/files/>)
- Incluir los archivos de la librería dentro del proyecto en Arduino.
- Usar el código ejemplo (ReadDemo)-Cargar en Arduino
- Para leer las áreas o escribir en ellas tener en cuenta la siguiente tabla:

Area table

	Value	Mean
<b>S7AreaPE</b>	0x81	Process Inputs.
<b>S7AreaPA</b>	0x82	Process Outputs.
<b>S7AreaMK</b>	0x83	Merkers.
<b>S7AreaDB</b>	0x84	DB
<b>S7AreaCT</b>	0x1C	Counters.
<b>S7AreaTM</b>	0x1D	Timers

Fig.2. Áreas de memoria que se pueden acceder con protocolo S7. Tomado de <http://settimino.sourceforge.net/>

- Tener en cuenta que el protocolo S7 es Big Endian, es decir, los bytes son leídos del mas significativo al menos significativo, mientras Arduino es Little Endian (Menos significativo a más significativo), entonces si tenemos una trama como:

**0x74** 0x72 0x65 0x63 0x65 0x0d **0x0a** (Big Endian)-> 0x74 sería el MSB y 0x0a el LSB (Big Endian-Protocolo S7)

**0x0a** 0x0d 0x65 0x63 0x65 0x72 **0x74**-> 0x0a sería el MSB y 0x74 el LSB (Little Endian-Arduino)

- Integrar el código dentro de la aplicación industrial que se requiera.
- Tener en cuenta lo siguiente el mapeo de memoria del PLCs7-1200 cuando se quiera acceder a bloques de memoria en Entradas, salidas o bits de memoria:

																								BYTE							
																WORD															
DWORD																															
x31	x30	x29	x28	x27	x26	x25	x24	x23	x22	x21	x20	x19	x18	x17	x16	x15	x14	x13	x12	x11	x10	x9	x8	x7	x6	x5	x4	x3	x2	x1	x0
b3								b2								b1								b0							
w1																w0															

Fig.3. Distribución de memoria PLC S7-1200.

## 4. Funciones para usar con S7 con *Settimino*

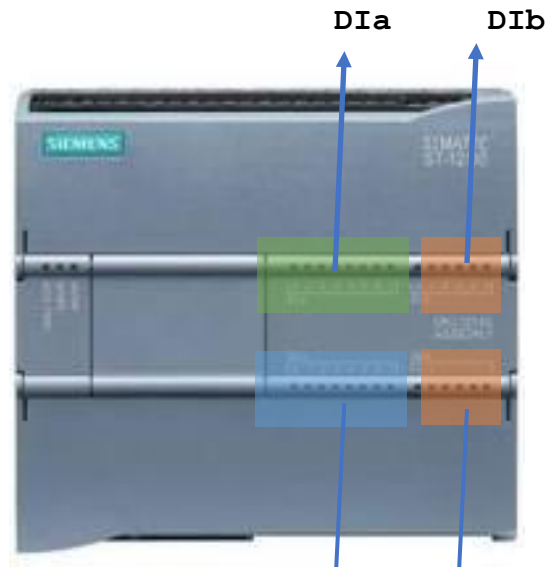
```
int ReadArea(int Area, uint16_t DBNumber, uint16_t Start,  
uint16_t Amount, void *pUsrData);
```

```
int WriteArea(int Area, uint16_t DBNumber, uint16_t Start,  
int Amount, void *pUsrData);
```

Nota: 1 Para escritura (WriteArea) el DBnumber es un área de memoria para las salidas (Coils-Qx.x), por ejemplo, el área DB0 con start 0 apuntará a las salidas DQa, mientras si DBnumber es 0 con start 1 apuntará a las salidas DQb.

Nota 2: Para la lectura de entradas, para leer el juego de entradas Dia y DIb se apunta al DBnumber 0 y se lee el número de entradas desde la posición start (0) a la cantidad de datos del buffer, por ejm, 10 bytes.

Entradas:



Salidas:

DQa DQb