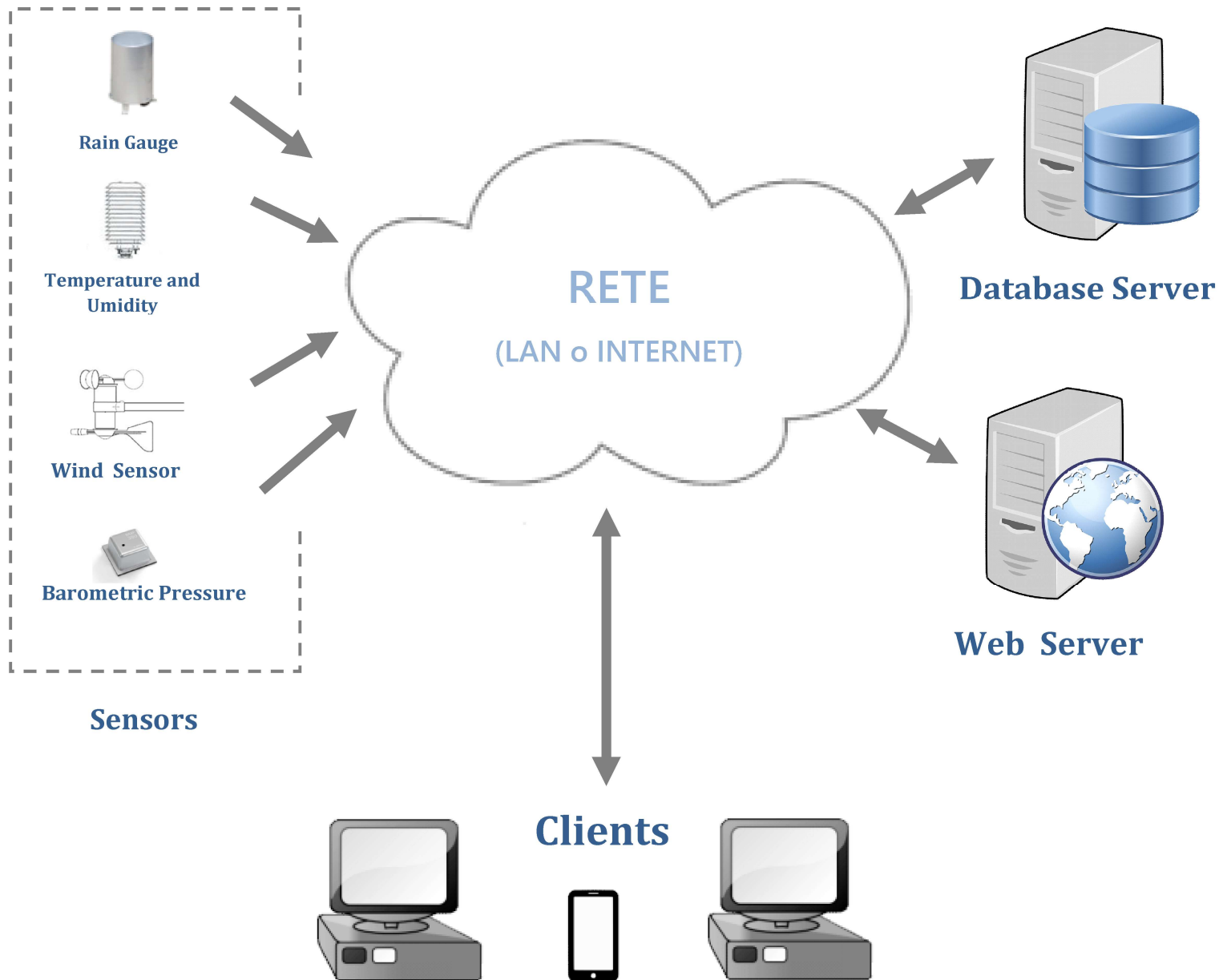
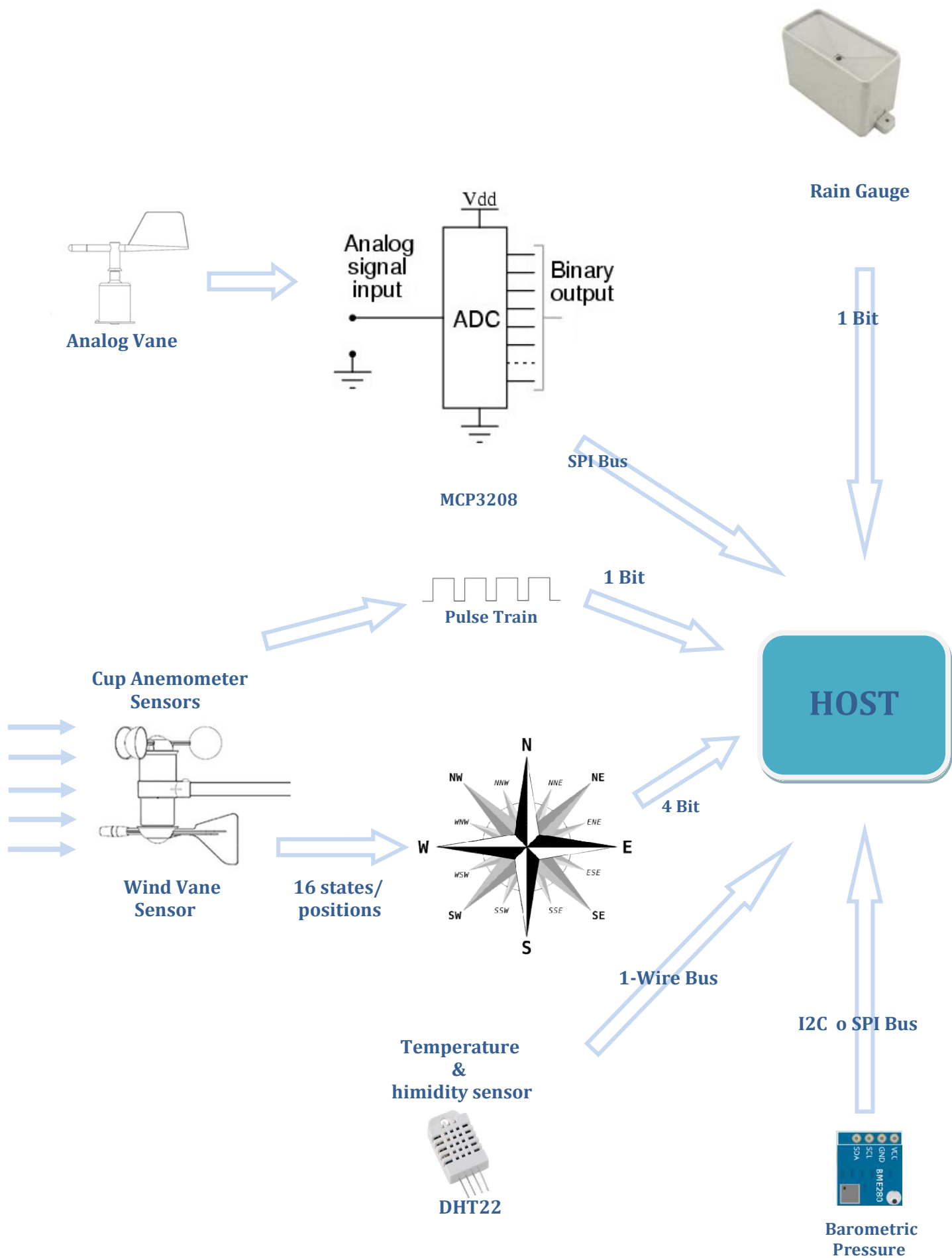
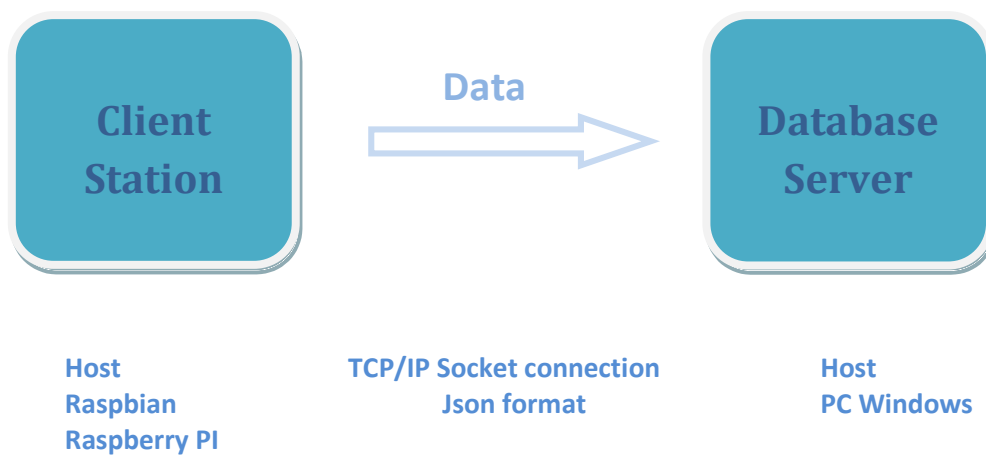


Weather Station



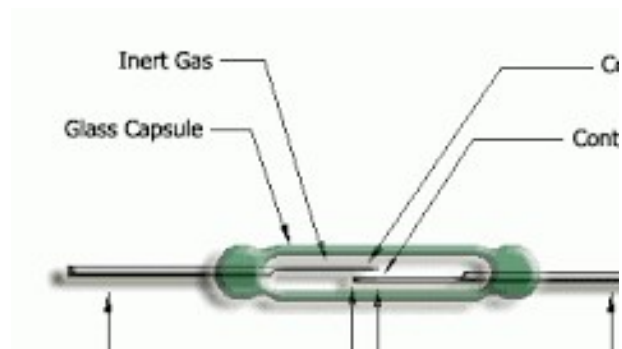
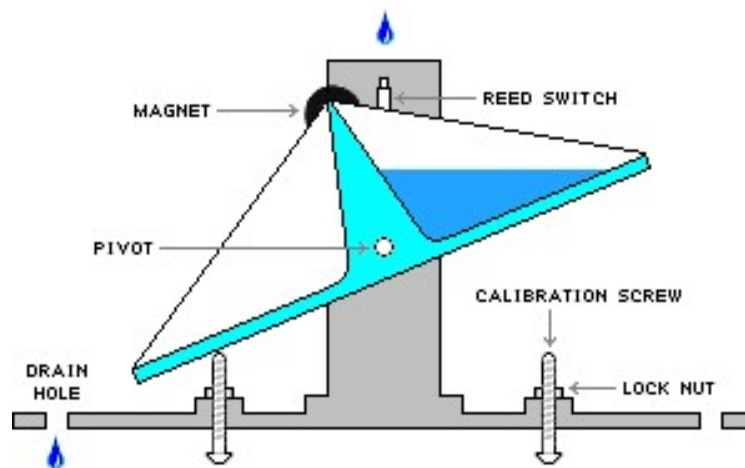




```
{  
  "W/m²":0.87,  
  "Km/h": 18.5,  
  "WD" : "SSW",  
  "°C":25.4,  
  "%RH": 73,  
  "NomeFile": "Foto.jpg",  
  "N_Byte": 8741285  
}
```

```
[  
  {"id": 2, "Dato": 0.87},  
  {"id": 3, "Dato": 18.5},  
  {"id": 1, "Dato": 9},  
  {"id": 14, "Dato": 25.4},  
  {"id": 8, "Dato": 73},  
  {"id": 21, "Dato": 8741285}  
]
```

Rain Gauge



La codifica di Gray

È un **codice numerico binario** che prende anche il nome di **codice ciclico** in quanto due cifre successive **differiscono solamente di 1 bit**: si dice che hanno **distanza unitaria** (distanza 1).

→ Distanza

In un codice con **distanza** si intende il numero di bit che "mutano" tra due configurazioni adiacenti.

Questo codice è stato progettato e brevettato nel 1953 nei **laboratori Bell** dal ricercatore **Frank Gray** per poterlo utilizzare nell'acquisizione di lettura di posizione di particolari **dispositivi elettronici** (come gli **encoder di posizione** utilizzati nei regolatori di volume digitali degli impianti hi-fi).

Il termine **encoder** è utilizzato in elettronica per indicare un dispositivo elettromeccanico che esegue la funzione di **trasduttore di posizione angolare**: esso converte la posizione angolare del suo asse rotante in numerici digitali che indicano il valore dell'angolo (chiamato **resolver**).

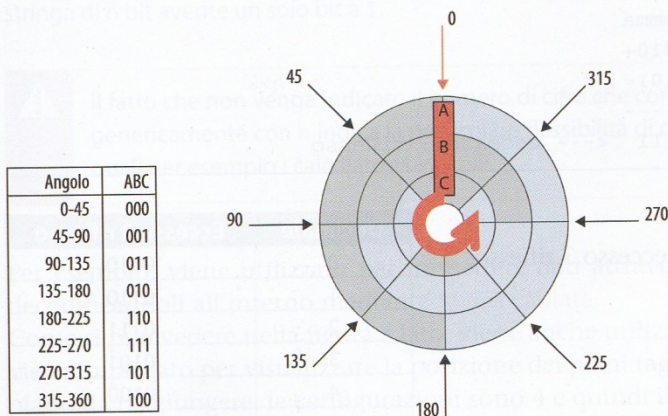
Esistono principalmente tre tipi di **trasduttori angolari** con uscita digitale:

- l'encoder incrementale;
- encoder tachimetrico;
- encoder assoluto.



A causa delle tolleranze meccaniche è improbabile che due o più bit di una cifra possano commutare esattamente nello stesso istante; viene quindi a crearsi un periodo intermedio in cui è codificato un **valore indesiderato**.

Negli **encoder** che utilizzano questo codice, il passaggio da un valore al successivo (o al precedente) comporta la commutazione di **un unico circuito**, eliminando ogni possibile valore equivoco.



Per ottenere il **codice di Gray** è possibile ricorrere alla **regola della specularità**: a partire dalla combinazione (0 1) è possibile costruire il codice per successive operazioni speculari, aggiungendo poi uno 0 per ogni cifra al di sopra della linea di specularità e un 1, sempre per ogni cifra, nella parte inferiore: nella tabella a lato abbiamo un esempio di costruzione del codice Gray a 4 bit. Queste cifre vanno man mano aggiunte nella parte sinistra del numero che si ottiene.

Nel passare da una parola a quella successiva cambia un solo bit, vengono minimizzati gli errori nel passaggio da uno stato al successivo e aumenta la velocità dell'**ALU**.

1 Bit	2 Bit	3 Bit	4 Bit
0	0 0	0 0 0	0 0 0 0
1	0 1	0 0 1	0 0 0 1
	—	0 1 1	0 0 1 1
	1 1	0 1 0	0 0 1 0
	1 0	—	0 1 1 0
		1 1 0	0 1 1 1
		1 1 1	0 1 0 1
		1 0 1	0 1 0 0
		1 0 0	—
			1 1 0 0
			1 1 0 1
			1 1 1 1
			1 1 1 0
			1 0 1 0
			1 0 1 1

Decimal Number	4 bit Binary Number <u>ABCD</u>	4 bit Gray Code <u>G₁G₂G₃G₄</u>
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

No!



Si!

