

## Interfaccia Sensori

<b>Autore</b>	Francesco D'Amore
<b>Data di creazione</b>	25-Ottobre-2023
<b>Ultima revisione</b>	25-Ottobre-2023
<b>Titolo</b>	D5.3 - Interfaccia Sensori
<b>Soggetto</b>	WP5 - Infrastruttura ICT
<b>Stato</b>	Completato
<b>Editore</b>	CNR-IIA
<b>Tipo</b>	Deliverable
<b>Identificazione</b>	D5.3
<b>Descrizione</b>	
<b>Contributi</b>	Mariantonia Bencardino, Delia Evelina Bruno, Valentino Mannarino

### INDICE

[INDICE](#)

[INTRODUZIONE E FINALITÀ](#)

[ARCHITETTURA DELL'INTERFACCIA SENSORI](#)

[UNITÀ DI PROCESSAMENTO](#)

### INTRODUZIONE E FINALITÀ

I dati prodotti dalla piattaforma multi sensore sviluppata nel WP2 devono essere trasmessi in risorse computazionali per poter essere processati e analizzati. La trasmissione dei dati non è sempre possibile per mancanza o scarsità di rete dati e deve essere sempre presente l'opzione batch, cioè l'accumulo di dati in sistemi locali e la successiva trasmissione massiva. Nel deliverable D5.2 è stata descritta una libreria (IoTty) adatta alla trasmissione in quasi tempo reale tramite protocolli IoT. IoTty è stata sviluppata per il progetto ARMONIA al fine di collegare il dispositivo prototipale per la produzione del dato (si vedano i risultati del WP2) con i sistemi di immagazzinamento e processamento in cloud: in questo deliverable invece verrà descritta l'interfaccia con i sensori, cioè di quei sistemi che raccolgono i segnali in corrente e la modificano in informazioni adatte ad essere eventualmente trasmesse con IoTty.

### ARCHITETTURA DELL'INTERFACCIA SENSORI

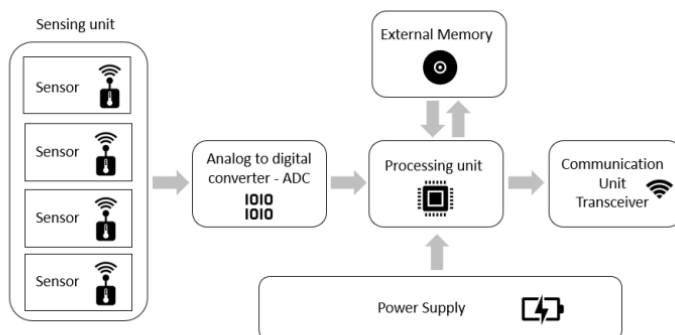
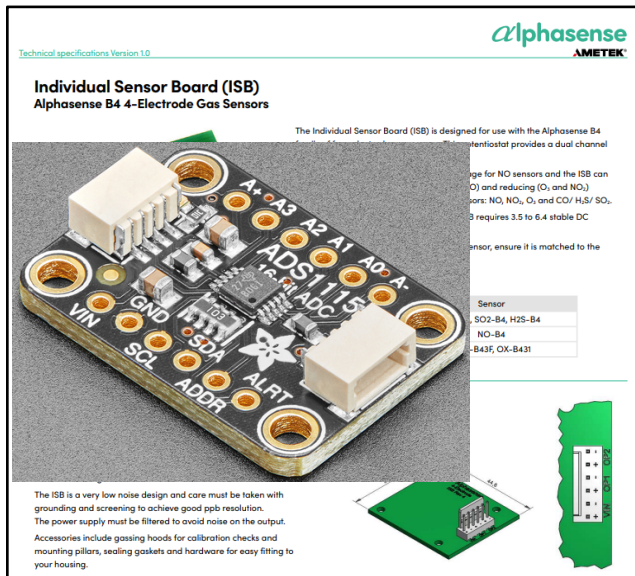


Figure 1. Sensor node architecture.

In Figura 1 si illustra schematicamente l'architettura adottata dall'interfaccia. I sensori sono interfacciati con delle schede proprietarie Alphasense (ISB - si veda figura) che prendono il segnale dei sensori in corrente e lo convertono in un sensore in tensione.

I segnali in corrente provenienti dai sensori inseriti nella piastra sensori (vedi D2.2), vengono convertiti in segnali in tensione dalle schede ISB Alphasense (vedi figura di fianco). Ogni sensori con uscita in corrente è accoppiato con la corrispondente scheda ISB. Il sensore di temperatura, pressione e umidità (PHRT) produce invece i dati direttamente in formato digitale attraverso una interfaccia i2c (Inter Integrated Circuit). Allo stesso modo, il sensore di particolato (OPC) espone dati attraverso un'interfaccia seriale. Per tutti gli altri è stato necessario raccogliere il segnale in tensione e usare un convertitore analogico digitale, come mostrato in Figura 1. A tale scopo è stato



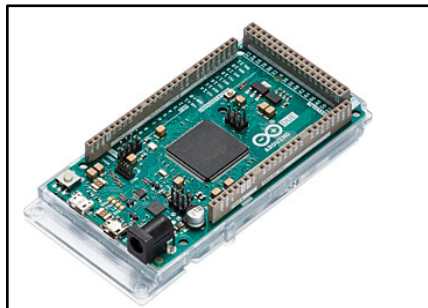
utilizzato il convertitore analogico digitale ADS115. Per i riferimenti ai sensori scelti per il progetto ARMONIA si veda il deliverable D2.1.

Una coppia del convertitore ADS115 è stata integrata in una scheda realizzata in laboratorio al fine di supportare i segnali provenienti dai sei sensori (nella configurazione completa) installabili nella piattaforma del prototipo. In Figura 1 questa scheda rappresenta il componente **Analog to Digital Converter - ADC**. I segnali in uscita da questa scheda, integrati con i segnali già digitali provenienti dai sensori con interfaccia seriale (OPC e PRHT) vengono

successivamente forniti al componente che sempre in Figura 1 viene identificato come **Processing Unit**. Il disegno prototipale prevede come Processing Unit una Raspberry PI (vedi deliverable D2.1), che consente di preprocessare il dato digitale prima di immagazzinarlo e inviarlo verso sistemi di processamento in cloud. Al fine di velocizzare lo sviluppo del prototipo, si è scelto in fase di realizzazione di utilizzare un microcontrollore **Arduino Due** che consente l'interfacciamento più agevole con i flussi seriali e uno sviluppo iterativo del prototipo. In una potenziale successiva fase di standardizzazione e consolidamento del prototipo, Arduino Due potrebbe essere sostituito o affiancato con una scheda di tipo Raspberry PI, come da progetto. L'utilizzo in fase prototipale di Arduino Due ha reso necessario l'inserimento di un sistema di storage dei dati esterno (chiamato External Memory in Figura 1) e di un clock esterno per gestire il tempo di sistema.

## UNITÀ DI PROCESSAMENTO

Come accennato nel paragrafo precedente, l'unità di processamento scelta per il prototipo è stato Arduino Due, sistema che consente una veloce prototipizzazione e un agevole interfacciamento con i convertitori ADC e, in cascata, con i sensori della piastra.



Un microcontrollore come Arduino DUE essenzialmente esegue in continuo, in un loop ad una certa frequenza, del codice che nel nostro caso si occupa di interrogare i sensori. Ad ogni iterazione il sistema assume il tempo da inserire nella lettura puntuale, interrogando un integrato collegato al microcontrollore che fornisce il clock, e immagazzina il dato come riga appesa ad un file salvato su una scheda micro SD.

Per i lettori interessati ai dettagli implementativi del ciclo di lettura, riportiamo alcuni estratti del codice sorgente. Innanzitutto il sistema viene inizializzato importando gli strumenti che servono alle letture strumentali.

```
#include <SD.h>
#include <Wire.h>
#include <Adafruit_ADS1X15.h>
#include <Arduino.h>
#include <Adafruit_SHT31.h> //Library for SHT31 - Temperature & humidity

Adafruit_ADS1115 ads1115_1; // Construct an ads1115 1
Adafruit_ADS1115 ads1115_2; // Construct an ads1115 2
Adafruit_SHT31 sht31 = Adafruit_SHT31(); //Creating an instance for SHT31
```

Si possono notare dal codice sopra i comandi necessari per interagire con i due convertitori ADS115. Successivamente è necessario configurare i sistemi attraverso una funzione di configurazione, riportata sotto.

```
void setup(void)
{
  myFile.println("SHT31 test");
  if (!sht31.begin(0x44)) { // Set to 0x45 for alternate i2c addr
    myFile.println("Couldn't find SHT31");
    while (1) delay(1);
  }

  ads1115_1.begin(0x48); // Initialize ads1115 at address 0x49
  ads1115_2.begin(0x49); // Initialize ads1115 at address 0x49

  Serial.begin(9600);
  Serial.print("Initializing SD card...");
  // On the Ethernet Shield, CS is pin 4. It's set as an output by default.
  // Note that even if it's not used as the CS pin, the hardware SS pin
  // (10 on most Arduino boards, 53 on the Mega) must be left as an output
  // or the SD library functions will not work.
  pinMode(10, OUTPUT);

  if (!SD.begin(10)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
}

}
```

Nella funzione di configurazione vengono impostati parametri necessari ad individuare le risorse dati (i convertitori ADC, ad esempio) ed a configurare i file nella scheda SD dove verranno conservate le informazioni. Finalmente viene fatto partire il loop che caratterizza il funzionamento di un microcontrollore. Nel loop che si riporta sotto vengono lette le informazioni dai sensori per tramite dei convertitori ADC per poi salvare i dati nella scheda SD precedentemente configurata.

```
void loop(void)
{
  int16_t op1_adc_CO, op2_adc_CO, op1_adc_NO2, op2_adc_NO2, op1_adc_OX, op2_adc_OX, op1_adc_SO2, op2_adc_SO2;
  float op1_CO, op2_CO, op1_NO2, op2_NO2, op1_OX, op2_OX, op1_SO2, op2_SO2;
  n++;
  //Considering sampling rate of 1 sample/2 minutes:
  //1 hour => n<=30
  //1 day=> n<=720
  //3 days=> n <=2160

  //CO SENSOR
  //Reads information from HE
  op1_adc_CO = ads1115_1.readADC_SingleEnded(0);
  //Converts ADC reading (raw counts) into volts
  op1_CO=ads1115_1.computeVolts(op1_adc_CO);
  //Reads information from HE
  op2_adc_CO = ads1115_1.readADC_SingleEnded(1);
  //Converts ADC reading (raw counts) into volts
  op2_CO=ads1115_1.computeVolts(op2_adc_CO);
```

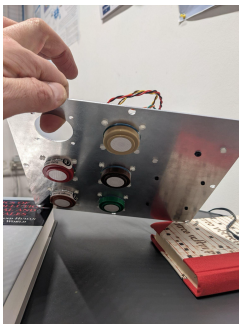
```
Float t = sht31.readTemperature();
Float h = sht31.readHumidity();
myFile = SD.open("test.txt", FILE_WRITE);
if (myFile){
  //Serial.println("Writing to test.txt...");
  myFile.print(n);myFile.print(",");
  //CO SENSOR
  //myFile.print(op1_adc_CO); myFile.print(","); //bits
  myFile.print(op1_CO * 1000,4); myFile.print(",");
  //myFile.print(op2_adc_CO); myFile.print(","); //bits
  myFile.print(op2_CO * 1000,4);myFile.print(",");
  //NO2 SENSOR
  //myFile.print(op1_adc_NO2); myFile.print(","); //bits
  myFile.print(op1_NO2 * 1000,4); myFile.print(",");
  //myFile.print(op2_adc_NO2); myFile.print(","); //bits
  myFile.print(op2_NO2 * 1000,4);myFile.print(",");
```

## REALIZZAZIONE

In questo paragrafo finale vengono mostrate alcune immagini del sistema descritto precedentemente e realizzato nella piastra sensori descritta nei deliverables afferenti il WP2.



I sensori sono alloggiati nella piastra attraverso gli ISB Alphasense descritti sopra che ne estraggono i segnali in corrente. La parte sensore risulta invece esposta come mostrato in figura centrale. La piastra di alloggiamento è estraibile al fine di consentire l'agevole gestione del dispositivo in fase di sviluppo e testing in laboratorio. Mentre quindi la zona inferiore viene esposta per la misurazione dei fenomeni di interesse, internamente il box protegge i dispositivi elettronici descritti sopra. Il risultato finale è mostrato nella terza figura sotto, dove tutti i componenti sono montati e interfacciati tra loro. Si possono notare i due ADS montati nell'integrato che li contiene, collegandoli direttamente alle schede ISB di ogni sensore che necessita una conversione digitale del segnale in tensione. Gli ADS sono poi collegati all'unità di processamento rappresentata dalla scheda Arduino Due anch'essa presente in figura. Il sistema si alimenta con una fonte a 12 Volt.



In figura l'alimentazione è fornita da un cavo posto nell'anima del palo di supporto, collegato al pannello fotovoltaico posto in cima ad esso.

