Solar Boat Celka

Data Logging and Visualization

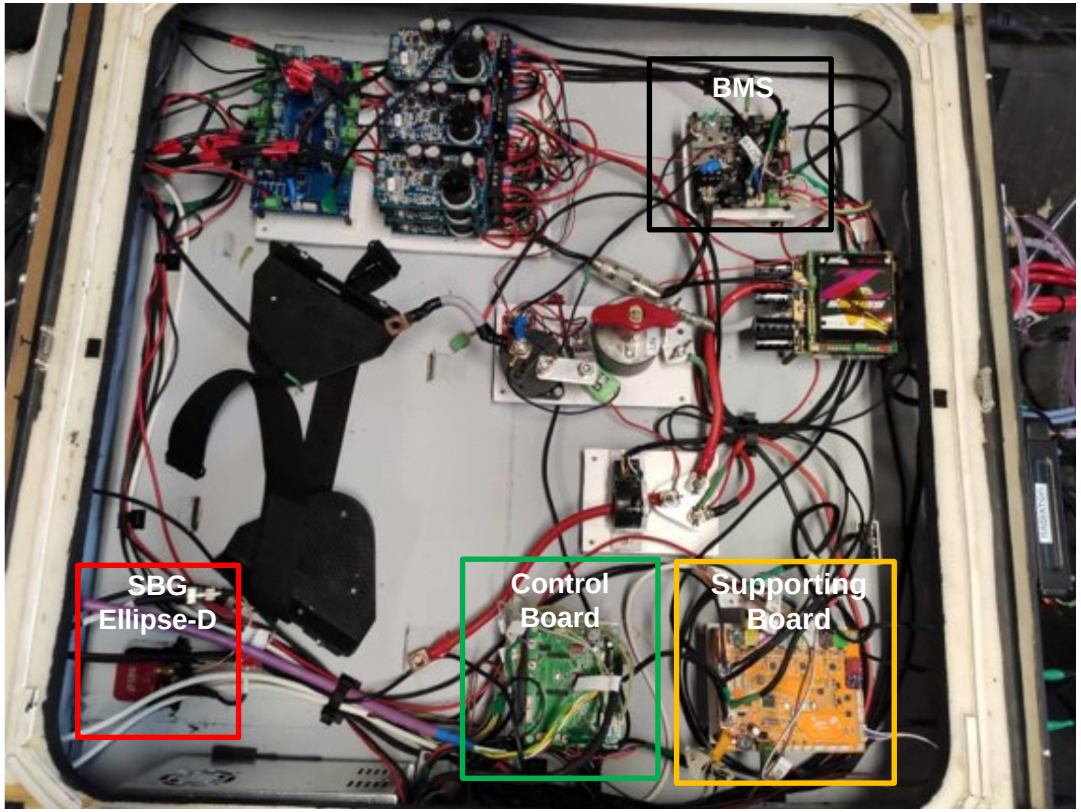
Project Objectives

1. Log data from subsystems of Solar Boat Celka using CSS CL2000 CAN logger.
2. Decode the logged data using a dbc file to extract meaningful information and signals.
3. Create a CSV file from the decoded data for seamless integration with InfluxDB.
4. Establish a connection between InfluxDB and Grafana for efficient data storage and retrieval.
5. Visualize the logged data in Grafana to gain insights, analyze trends, and support decision-making processes.

Logging Data from Subsystems

1. **Battery Management System**
   - manages and monitors the performance and health of the battery system, ensuring optimal utilization and safety.
2. **Control Board**
   - monitors motor parameters, controls motor operations, and regulates the servomechanism of hydrofoils.
3. **Supporting Board**
   - monitors and controls the parameters of various devices such as pumps, fans, and other supporting equipment.
4. **User Input Board**
   - provides an interface for users to input commands and interact with the system, facilitating user control and interaction.
5. **SBG Ellipse-D - Inertial Navigation System**
   - integrates sensors and algorithms for accurate positioning, orientation, and navigation using IMU, EKF, and dual antenna GNSS technology.

Subsystems of Celka



CAN Bus Logger

The CSS CL2000 CAN Logger is a industrial device designed for capturing and logging Controller Area Network (CAN) data.

Logging Features:

- Logs raw CAN data, including timestamps, message IDs, and data payloads in simple CSV style format.
- Configurable logging parameters such as sample rate, message filtering, and storage options.
- Stores logged data on a removable SD card for convenient data retrieval.

Sample Logs:

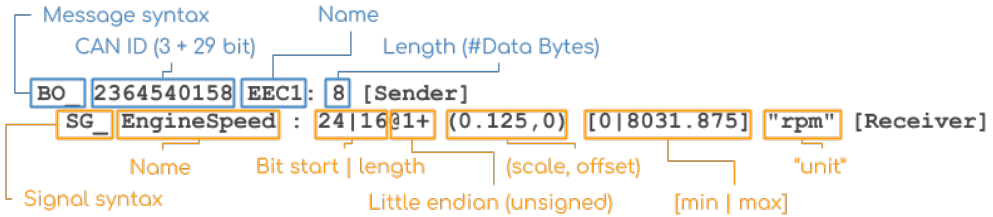
```
# Logger type: CL2000
# HW rev: 8.1x
# FW rev: 5.85
# Logger ID: CAN_logger_SBT
# Session No.: 177
# Split No.: 10
# Time: 20230523T065634
# Value separator: ";"
# Time format: 3
# Time separator: ":"
# Time separator ms: ","
# Date separator: ""
# Time and date separator: "T"
# Bit-rate: 125000
# Silent mode: true
# Cyclic mode: false
Timestamp;ID;Data
06:56:33,986;22;7285bf88e71e8afe
06:56:33,987;23;4a1191101e110000
06:56:33,988;0;ff01000000000000
06:56:33,989;1;70fd7cfd00000000
```

Decoding Data with DBC File

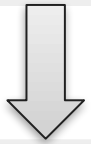
CAN DBC (CAN database) file is a text file that stores essential information for decoding raw CAN bus data into meaningful 'physical values'.

Structure of a DBC file:

- Messages
- Signals




```
CAN ID    Data bytes
0CF00400  FF FF FF 68 13 FF FF FF
```



```
Message  Signal       Value  Unit
EEC1     EngineSpeed  621    rpm
```

Celka DBC file

CAN messages

	Name	CAN ID (HEX)	Type	DLC	Comment
○	EKF_VEL	1F	Standard	6	
○	SHIP_MOTION_0	20	Standard	6	
○	MOTOR_STATUS_ID	21	Standard	8	
●	BMS_DATA_ID	22	Standard	8	
○	MAIN_BOX_TEMP_ID	23	Standard	8	

CAN signals (BMS_DATA_ID)

	Name	Type	Order	Mode	Start	Length	Factor	Offset	Min	Max	Unit	Comment
●	MAX_CELL_VOLTAGE	Unsigned	Intel	Signal	0	16	0.0001	0	0	5	V	
○	MIN_CELL_VOLTAGE	Unsigned	Intel	Signal	16	16	0.0001	0	0	5	V	
○	DISCHARGE_CURRENT	Unsigned	Intel	Signal	32	16	0.01	0	-20	500	A	
○	CHARGE_CURRENT	Unsigned	Intel	Signal	48	16	0.01	0	0	100	A	

CAN signal preview ☐

DBC preview (BMS_DATA_ID)

```
BO_ 34 BMS_DATA_ID: 8 CELKA
 SG_ MAX_CELL_VOLTAGE : 0|16@1+ (0.0001,0) [0|5] "V" Vector__XXX
 SG_ MIN_CELL_VOLTAGE : 16|16@1+ (0.0001,0) [0|5] "V" Vector__XXX
 SG_ DISCHARGE_CURRENT : 32|16@1+ (0.01,0) [-20|500] "A" Vector__XXX
 SG_ CHARGE_CURRENT : 48|16@1+ (0.01,0) [0|100] "A" Vector__XXX
```

Creating CSV File for InfluxDB

Extended Annotated CSV File

- file format used for storing data that will be integrated with InfluxDB,
- provides additional metadata and annotations specifying how CSV data should be converted to line protocol.

Line protocol

- text-based format used to insert data into the InfluxDB,
- defines the structure and syntax for representing data points in InfluxDB.

```
17  Timestamp;ID;Data
18  06:56:33,986;22;7285bf88e71e8afe
19  06:56:33,987;23;4a1191101e110000
20  06:56:33,988;0;ff01000000000000
21  06:56:33,989;1;70fd7cfd00000000
```

```
1   #group,false,false,false,true,true,false,false
2   #datatype,string,long,dateTime:RFC3339,string,string,double,string
3   #default,_result,,,,,,
4   ,result,table,_time,_measurement,_field,_value,symbol
5   ,,0,2023-06-03T06:56:33.986Z,BMS_DATA_ID,MAX_CELL_VOLTAGE,3.4162,V
6   ,,0,2023-06-03T06:56:33.986Z,BMS_DATA_ID,MIN_CELL_VOLTAGE,3.5007,V
7   ,,0,2023-06-03T06:56:33.986Z,BMS_DATA_ID,DISCHARGE_CURRENT,79.11,A
8   ,,0,2023-06-03T06:56:33.986Z,BMS_DATA_ID,CHARGE_CURRENT,651.62,A
9   ,,0,2023-06-03T06:56:33.987Z,MAIN_BOX_TEMP_ID,TEMP1,44.26,C
10  ,,0,2023-06-03T06:56:33.987Z,MAIN_BOX_TEMP_ID,TEMP2,42.41,C
11  ,,0,2023-06-03T06:56:33.987Z,MAIN_BOX_TEMP_ID,TEMP3,43.82,C
12  ,,0,2023-06-03T06:56:33.988Z,MOTOR_POWER,MAIN_MOTOR_POT,51.1,%
13  ,,0,2023-06-03T06:56:33.988Z,MOTOR_POWER,PADDLE_MOTOR_POT,0.0,%
14  ,,0,2023-06-03T06:56:33.989Z,HYDROFOIL_ANGLES,LEFT_ANGLE,-6.56,
15  ,,0,2023-06-03T06:56:33.989Z,HYDROFOIL_ANGLES,RIGHT_ANGLE,-6.44,
```

InfluxDB - Time-Series Database

InfluxDB is a high-performance, open-source time series database designed for collecting, storing, and analyzing time-stamped data.

Key Features:

- Time Series Data
- High Write and Query Performance
- Query Language: InfluxQL and Flux
- Data Retention Policies
- Integration

Use Cases:

- Internet of Things (IoT) applications
- DevOps and infrastructure monitoring
- Real-time analytics and anomaly detection

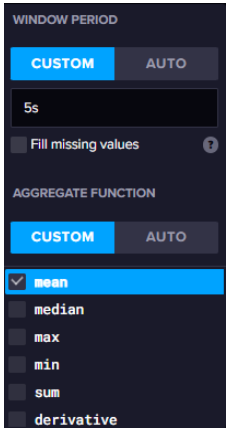
Uploading data to InfluxDB

To log in to the local server, users utilize the following URL: http://localhost:8086/.

- Create bucket in InfluxDB:
- The bucket is name "DA_" within the InfluxDB project.
- Create an API token:
- A token is create specifically for the Grafana integration "DA_".
- Import data from the prepared CSV file to created bucket.
- Example code:

This query retrieves data from the "DA" bucket, filters it based on measurement, field, and symbol criteria, and then applies the desired aggregation function, such as calculating the mean values within 5-second intervals.

```
from(bucket: "DA_")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "BMS_DATA_ID")
  |> filter(fn: (r) => r["_field"] == "MAX_CELL_VOLTAGE" or r["_field"] == "MIN_CELL_VOLTAGE")
  |> filter(fn: (r) => r["symbol"] == "V")
  |> aggregateWindow(every: 5s, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```



Grafana - Data Visualization and Monitoring

Grafana is a popular open-source visualization and analytics platform designed for monitoring, analyzing, and visualizing data.

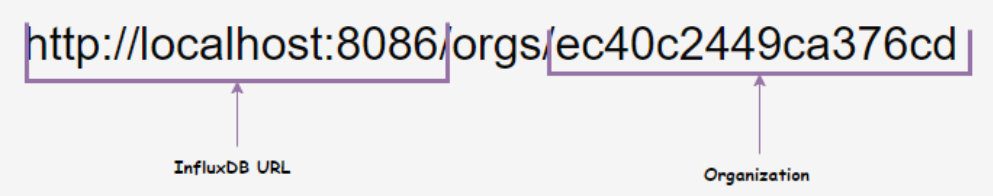
Key Features:

- Data Visualization
- Dashboard Creation
- Data Source Integration
- Alerting and Notifications

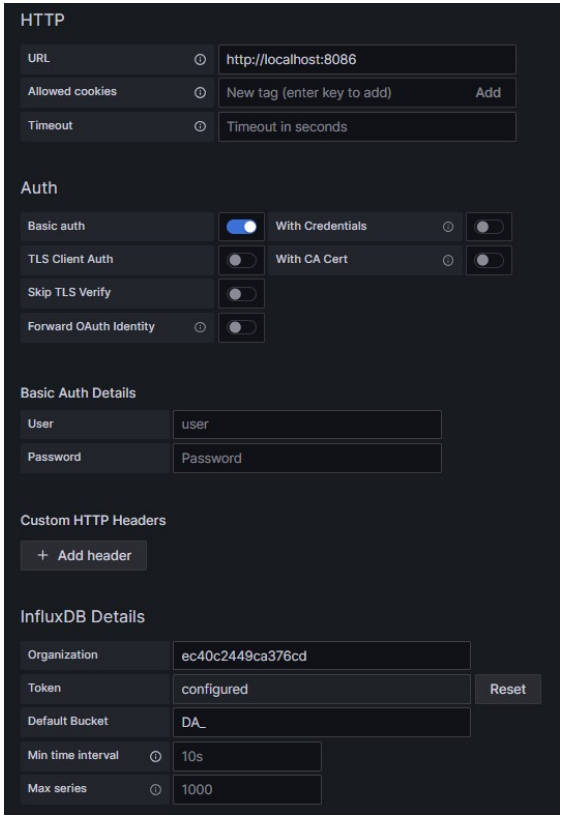
Use Cases:

- Real-time Monitoring
- DevOps and IT Operations
- Business Intelligence
- IoT Data Visualization

Connecting InfluxDB with Grafana

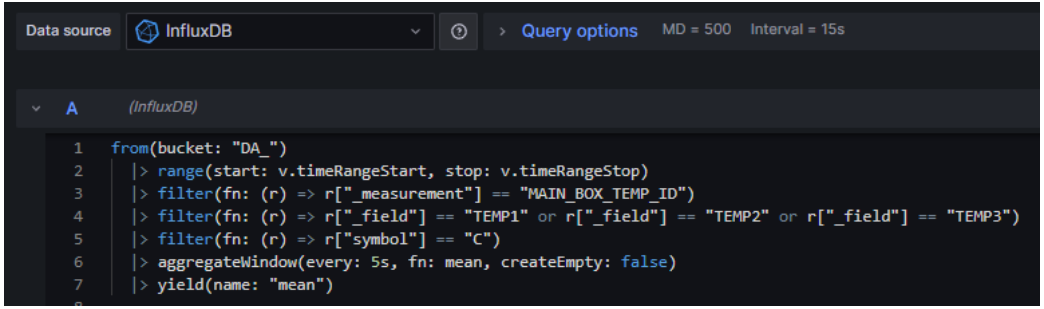


- Access the local Grafana server through the following URL: http://localhost:3000/.
- Set InfluxDB as the new data source
- Set "Query Language" to Flux
- Specify InfluxDB details
  - Default Bucket,
  - Token
  - InfluxDB URL



Visualizing Data with Grafana

- Three dashboards have been created for data visualization:
  - Celka Telemetry Basic Info
  - SGB Ellipse-D IMU
  - Voltages & currents
- For each dashboard, the corresponding time when the data was collected has been specified:
  - from 2023-06-03 08:06:21 to 2023-06-03 10:15:12
- The charts have been configured so that the position graph added to each dashboard shows the exact location where the data was collected.
- Flux queries generated via Query Builder tool in InfluxDB have been used to acquire data for visualization.
- Many charts from Time Series to GeoMap are available for data visualization.



Demonstration and Results

Conclusion

Successful achievements:

- Logged data from subsystems of Solar Boat Celka.
- Developed a custom script for:
  - decoding the logged data using a dbc file,
  - creating extended annotated CSV file for InfluxDB input.
- Established a connection between InfluxDB and Grafana.
- Visualized the logged data in Grafana.

Future plans:

- Migrate the system to a remote server for improved accessibility and scalability.
- Developing real-time telemetry of Celka boat.