A Cost-effective Automated Weather Reporting System AWRS for The Canadian Remote Northern Air Operators

Houssam Hammoudi *houssam@houssamhammoudi.com*
CEO, TradeSec Corp
Calgary, T3N 1T4, Canada

Abstract

Air transportation is essential for Canada's and US northern communities. It is the leading lifeline supplying fresh food, medicine, and other goods; providing Health care services; medical emergency evacuation; and supporting travel outside of the communities. In addition, air transportation is the only reliable year-round mode of transportation. However, the Canadian north and Alaska present significant operational challenges mainly due to inhospitable terrain, harsh conditions, extreme cold. The challenges are financial as well due to low passenger volumes and high operations costs. This research shows how northern Air operators can enhance flight safety using WX-Ready as an AWRS "Automated Weather Information System" to get vital weather information without investing in expensive commercially available ACARS systems (Government of Canada, 2017).

Keywords: Software Engineering, ACARS, Python, Linux, METAR, TAF, Aviation, Air Operators.

1. INTRODUCTION

Aviation is an essential lifeline of the world's economy; it supports more than 65.5 million jobs worldwide and enables $2.7 trillion in global GDP (Government of Canada, 2017). The aviation sector in Canada generates more than 632000 jobs and $49 billion to contribute to the Canadian GDP. Although Canada has 500 airports, only about half of these airports have scheduled commercial flights. However, most of these airports are small. From these 500 airports, only 117 are remote northern airports scattered across the vast Canadian north. In addition, many northern communities operate ice strips in the arctic region, which are not certified nor listed as airports by Transport Canada. (Government of Canada, 2017).

2. OBJECTIVES

The study aims to develop a Python/Linux Weather retrieval and display system that replaces the expensive commercially available ACARS systems and enhances flight safety without a significant investment. Small operators cannot afford a half-million dollars ACARS solution. Our objective is to fill this need and fix this operational problem in the industry by providing an affordable AWRS system that can enhance flight safety at minimal costs. In addition, we will leverage the GSM/Cellular network and its SMS capability to send and retrieve messages. We will also set a concept for future development using Iridium Satellite Short Burst Data SBD messaging.

3. LITERATURE REVIEW

Reviewing similar research papers, we found that most of the research papers focused on operations within commercial airports (Benjamin & Moninger, 2016). We, however, focused on northern operators operating in and from ice strips and small northern airports. We realized that most of the literature is only aimed at established airlines with big commercial jets; we couldn't find anything related to small operators, charter or commuter operators with aircrafts less than 19 passengers. Another observation was that none of the studies covered Medivac operators that are vital in Canada (International Civil Aviation Organization, 2010).

3.1 The Importance of Aviation Weather
The aviation weather systems have evolved tremendously since the 1980s; in addition to fixed weather observation stations, aircraft with their onboard sensors take atmospheric measurements and share that information with other aircraft and ground stations via the Aircraft Meteorological Data Relay (AMDAR). (Anaman et al., 2017; Federal Aviation Administration, 2009).

3.2 Important Weather Readings
Aviation weather information comes in the form of forecasts (Federal Aviation Administration, 2009) observations, and notices. We are going to focus only on the following:

- **METAR** – Meteorological Terminal Air Report
- **TAF** – Terminal Aerodrome Forecast
- **NOTAM** – Notice to airmen

4. PROJECT DESIGN AND METHODOLOGY
The methodology used in this study follows the software prototyping development model (Susanto et al., 2019) This model allows building, testing, and refining the software prototype based on end-user feedback until a good working version is accomplished.

4.1 The Requirements Gathering and Analysis Phase
The requirement gathering and analysis phase consisted of being on the field gathering all the observations made by the pilots, the dispatchers, and the company management. We witnessed the lack of weather information during the study during ferry flights over remote northern areas and transoceanic flights.

4.2 The Quick Design Phase
This section presents a quick and straightforward framework showing how the user interface will appear.

4.3 The Build Phase
This section produces a quick and simple design that we call an MVP a “Minimum Viable Product.” This version of the apparatus included only four features. This phase allowed us to gather valuable feedback about the product

4.4 Field Evaluation Phase
In this section, we describe how we used the apparatus in the field during transoceanic flights and flights conducted in the Arctic region. This phase was crucial to gathering valuable feedback from the pilots, allowing us to determine its strengths and weaknesses and the must-have features.

4.5 Refining The Prototype Phase
In this section, we measured the impact of the apparatus, we determined the nice-to-have features and the must-have features, and we were able to have an approved final prototype.

5. HARDWARE COMPONENTS
This section will describe in detail the different hardware components used in this project and explain their roles.

5.1 Single Board Computer
This apparatus combines software and a single board computer that needs to bring a small size factor, versatility, and upgrade either by expansion boards or hardware attached on top boards (HAT) and cost-effective. We found all these characteristics in the Raspberry Pi 3 Model B+ shown in figure 1 below.

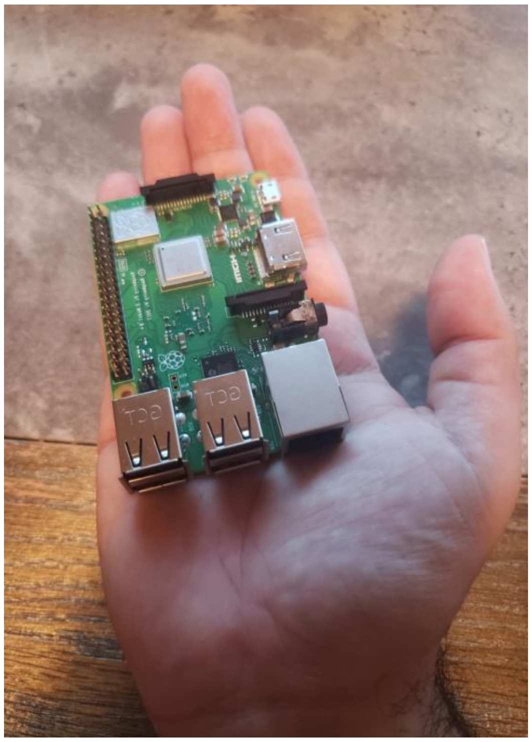


FIGURE 1: Raspberry Pi Model B+.

5.2 SIM800 GSM/GPRS HAT Expansion Board

This HAT board is explicitly made for the Raspberry Pi. It attaches on top of the Pi by connecting to the 40 GPIO pins shown in figure 2.

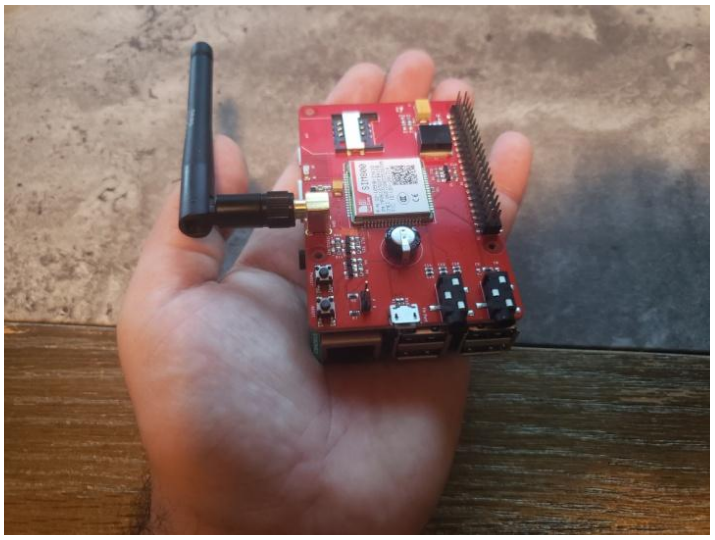


FIGURE 2: SIM800 HAT board connected to the Raspberry Pi.

5.3 TFT 3.5-inch LCD Touch Screen

The 3.5-inch LCD touch screen displays the weather data and allows users to make selections and input airport codes. The LCD screen connects to the raspberry pi via the 40 GPIO pins of the SIM800 board shown in figure 3.

FIGURE 3: TFT 3.5-inch LCD Touch Screen.

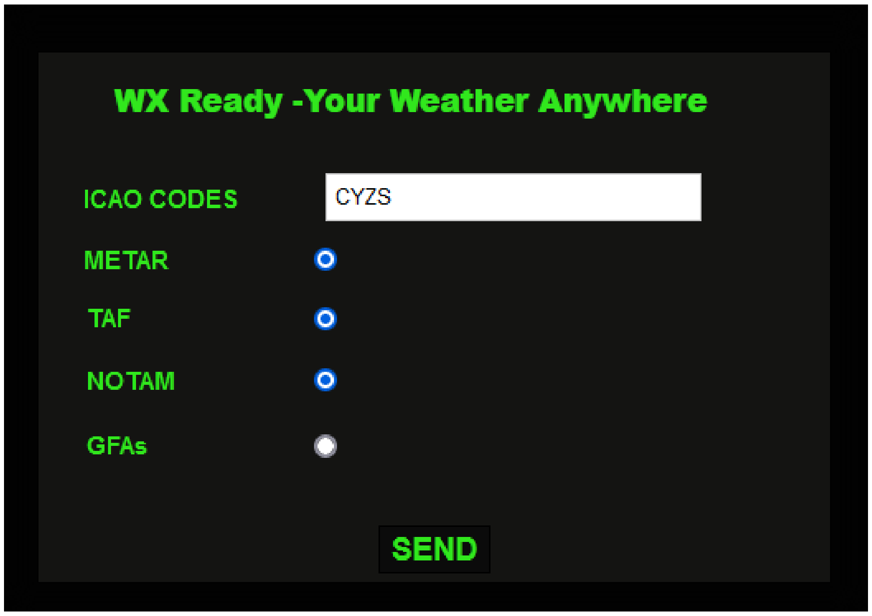
6. USER INTERFACE AND NAVIGATION

We chose to develop the Linux Debian-like OS called Raspbean because it provides a cost-effective, stable, and robust platform. We also decided to write the UI using the Tkinter Python framework due to its various GUI elements and ease of use.

6.1 Airport Selection

6.1.1 ICAO CODES UI Component

This section of the UI allows the pilot to enter all airports they wish to retrieve weather info. Commas can separate the airport codes. The crew members can select which information they want to receive by selecting the METAR, TAF, NOTAM, and GFAs as depicted in figure 4.



WX Ready -Your Weather Anywhere

ICAO CODES: CYZS

METAR [✓]

TAF [✓]

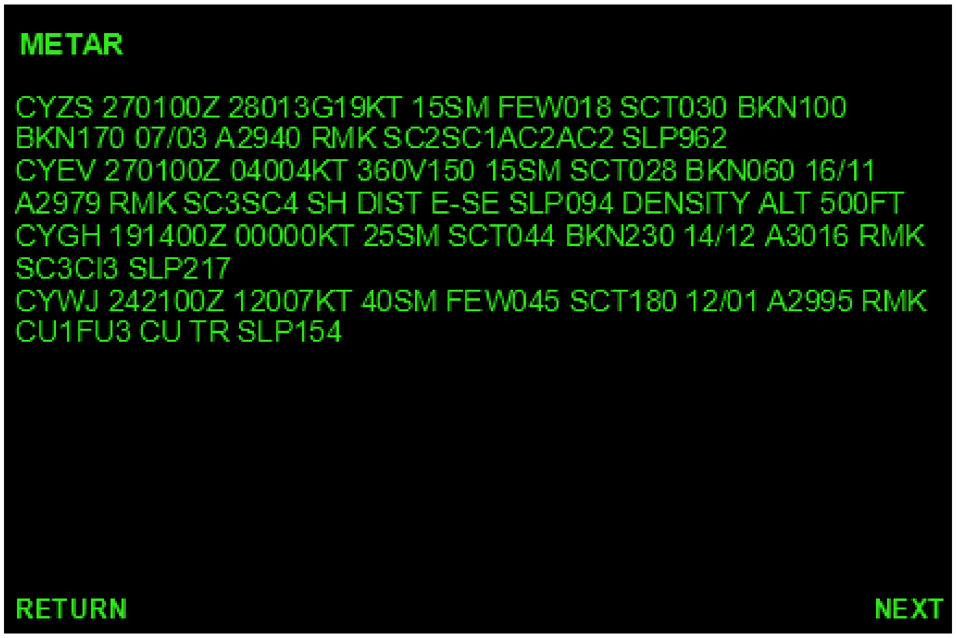
NOTAM [✓]

GFAs [ ]

FIGURE 4: The airport, METAR, TAF, and Notam selection.

6.1.2 METAR Weather Info Display UI Component

This screen depicted in figure 5 below shows the METAR information of every airport in the airport list.

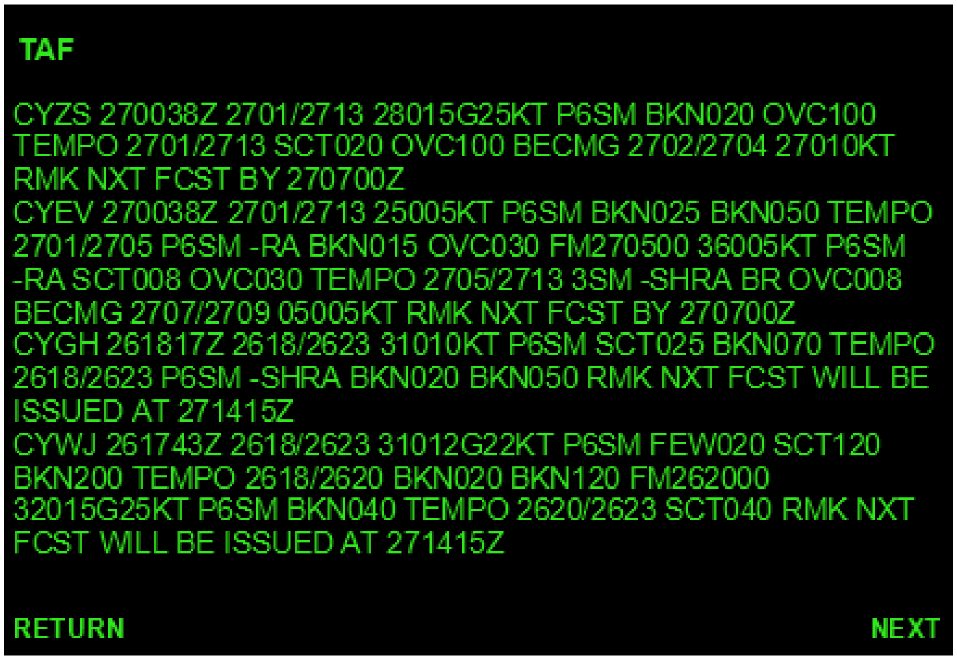


```
CYZS 270100Z 28013G19KT 15SM FEW018 SCT030 BKN100
BKN170 07/03 A2940 RMK SC2SC1AC2AC2 SLP962
CYEV 270100Z 04004KT 360V150 15SM SCT028 BKN060 16/11
A2979 RMK SC3SC4 SH DIST E-SE SLP094 DENSITY ALT 500FT
CYGH 191400Z 00000KT 25SM SCT044 BKN230 14/12 A3016 RMK
SC3CI3 SLP217
CYWJ 242100Z 12007KT 40SM FEW045 SCT180 12/01 A2995 RMK
CU1FU3 CU TR SLP154
```

FIGURE 5: METAR information displayed.

6.1.3 TAF Weather Info Display UI Component

The figure 6 below shows the METAR information of every airport in the airport list.

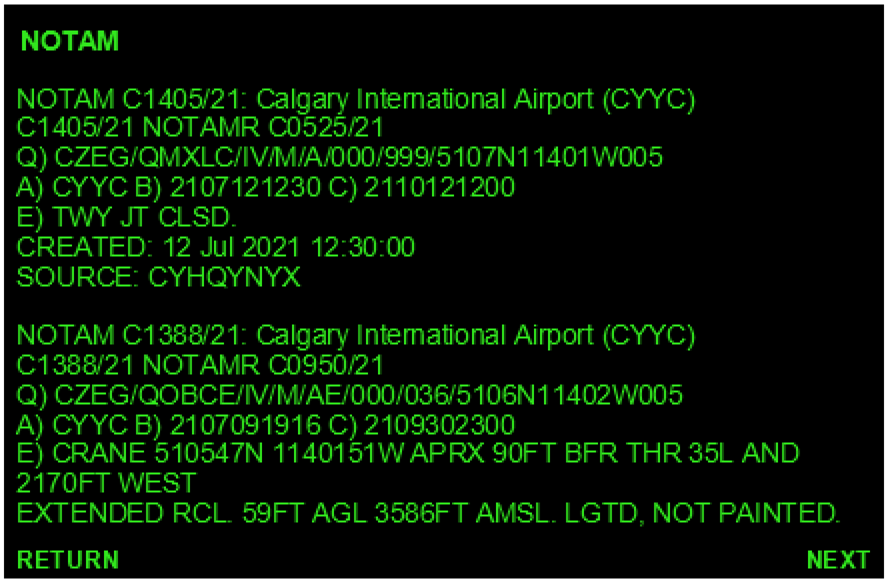


```
CYZS 270038Z 2701/2713 28015G25KT P6SM BKN020 OVC100
TEMPO 2701/2713 SCT020 OVC100 BECMG 2702/2704 27010KT
RMK NXT FCST BY 270700Z
CYEV 270038Z 2701/2713 25005KT P6SM BKN025 BKN050 TEMPO
2701/2705 P6SM -RA BKN015 OVC030 FM270500 36005KT P6SM
-RA SCT008 OVC030 TEMPO 2705/2713 3SM -SHRA BR OVC008
BECMG 2707/2709 05005KT RMK NXT FCST BY 270700Z
CYGH 261817Z 2618/2623 31010KT P6SM SCT025 BKN070 TEMPO
2618/2623 P6SM -SHRA BKN020 BKN050 RMK NXT FCST WILL BE
ISSUED AT 271415Z
CYWJ 261743Z 2618/2623 31012G22KT P6SM FEW020 SCT120
BKN200 TEMPO 2618/2620 BKN020 BKN120 FM262000
32015G25KT P6SM BKN040 TEMPO 2620/2623 SCT040 RMK NXT
FCST WILL BE ISSUED AT 271415Z
```

FIGURE 6: TAF information displayed.

6.1.4 NOTAM Weather Info Display UI Component

The figure 7 below shows the NOTAM information of every airport in the airport list.



```
NOTAM

NOTAM C1405/21: Calgary International Airport (CYYC)
C1405/21 NOTAMR C0525/21
Q) CZEG/QMXLC/IV/M/A/000/999/5107N11401W005
A) CYYC B) 2107121230 C) 2110121200
E) TWY JT CLSD.
CREATED: 12 Jul 2021 12:30:00
SOURCE: CYHQYNYX

NOTAM C1388/21: Calgary International Airport (CYYC)
C1388/21 NOTAMR C0950/21
Q) CZEG/QOBCE/IV/M/AE/000/036/5106N11402W005
A) CYYC B) 2107091916 C) 2109302300
E) CRANE 510547N 1140151W APRX 90FT BFR THR 35L AND
2170FT WEST
EXTENDED RCL. 59FT AGL 3586FT AMSL. LGTD, NOT PAINTED.
RETURN                                              NEXT
```

FIGURE 7: NOTAM information displayed.

7. SYSTEM ARCHITECTURE

The system architecture depicted in figure 8 is comprised of different components and services.

- Webserver containing the web service listener
- SMS 3rd party service API “Twilio.”
- SMS 3rd party webhook service “Twilio.”
- SIM800 receiving and sending SMS messages

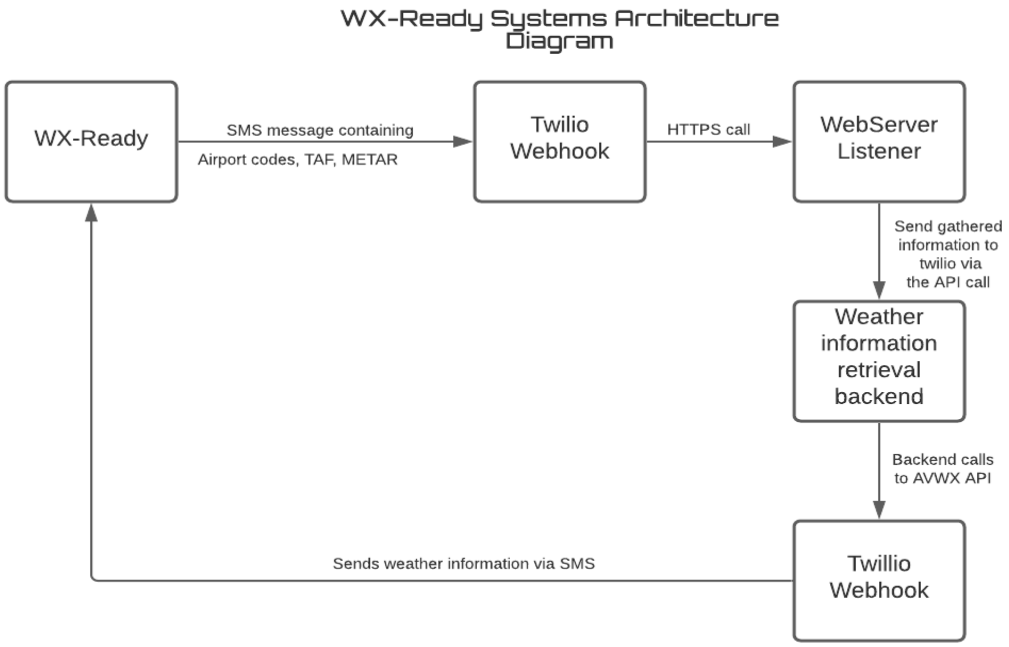

WX-Ready Systems Architecture Diagram
WX-Ready
SMS message containing
Airport codes, TAF, METAR
Twilio Webhook
HTTPS call
WebServer Listener
Send gathered information to twilio via the API call
Weather information retrieval backend
Backend calls to AVWX API
Twillio Webhook
Sends weather information via SMS

FIGURE 8: Systems architecture diagram.

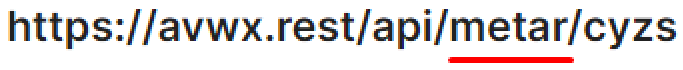
7.1 Webserver
The web server is an Apache server running on a Debian Linux OS. We chose Apache for its ease of use and stability. The webserver is hosted publicly and responds only to HTTPS calls for added security.

7.2 Twilio Messaging Service
Twilio is a cloud communication platform allowing developers to send calls, SMS, and MMS messages using their many web service APIs. Moreover, the inbound and outbound SMS message prices are minimal, costing only $0.0075 per message. This price point makes the WX-Ready very attractive and advantageous for northern operators who cannot afford a hundred thousand dollars ACARS system.

7.3 Twilio Webhook Service
Twilio is a cloud communication platform allowing developers to send calls, SMS, and MMS messages using their many web service API's.

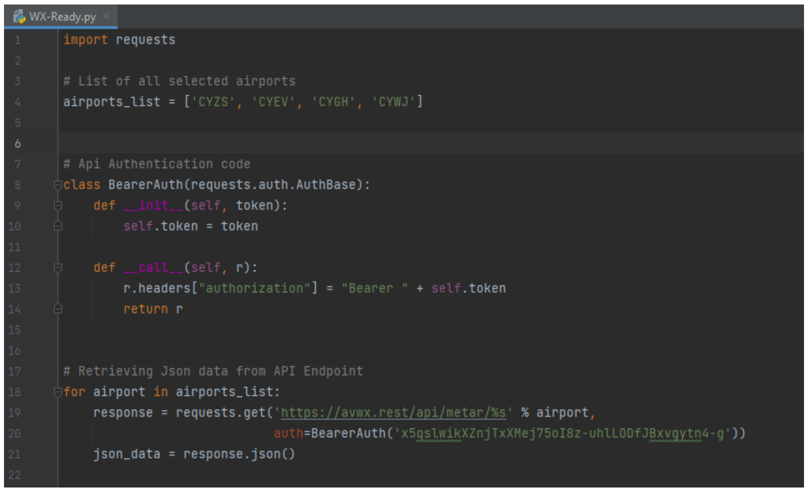
7.4 Twilio Webhook Service
Twilio is a cloud communication platform allowing developers to send calls, SMS, and MMS messages using their many web service API's.

8. WEATHER INFORMATION RETRIEVAL BACKEND
8.1 Backend METAR Info Retrieval "GET request."
The backend component is the code that runs to retrieve the weather observations of the specified airports. This backend code uses API calls to an aviation Weather REST API service. The METAR has its specific endpoint shown in figure 9 and figure 10.



https://avwx.rest/api/metar/cyzs

FIGURE 9: METAR API Endpoint

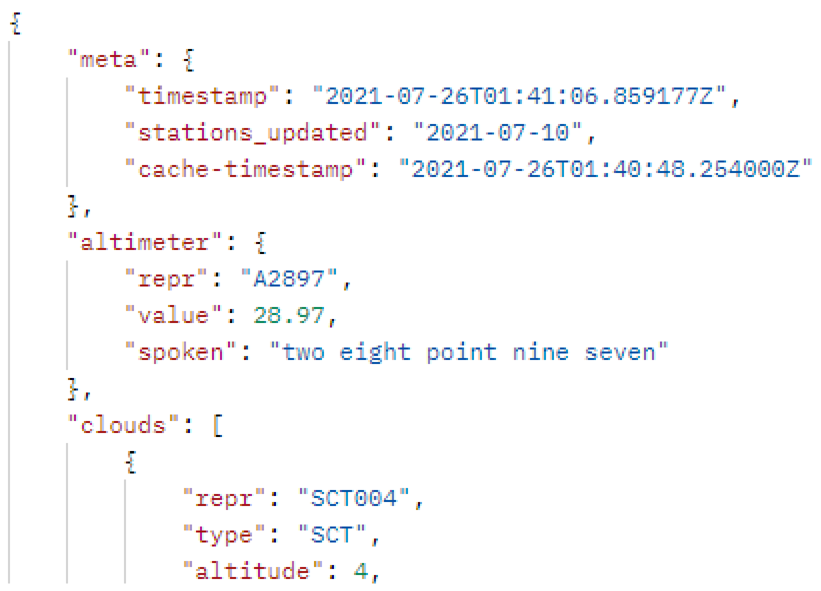


```python
import requests

# List of all selected airports
airports_list = ['CYZS', 'CYEV', 'CYGH', 'CYWJ']


# Api Authentication code
class BearerAuth(requests.auth.AuthBase):
    def __init__(self, token):
        self.token = token

    def __call__(self, r):
        r.headers["authorization"] = "Bearer " + self.token
        return r


# Retrieving Json data from API Endpoint
for airport in airports_list:
    response = requests.get('https://avwx.rest/api/metar/%s' % airport,
                            auth=BearerAuth('x5qslwikXZnjTxXMej75oI8z-uhlLODfJBxvgytn4-g'))
    json_data = response.json()
```

FIGURE 10: Python Code used to request weather data in.

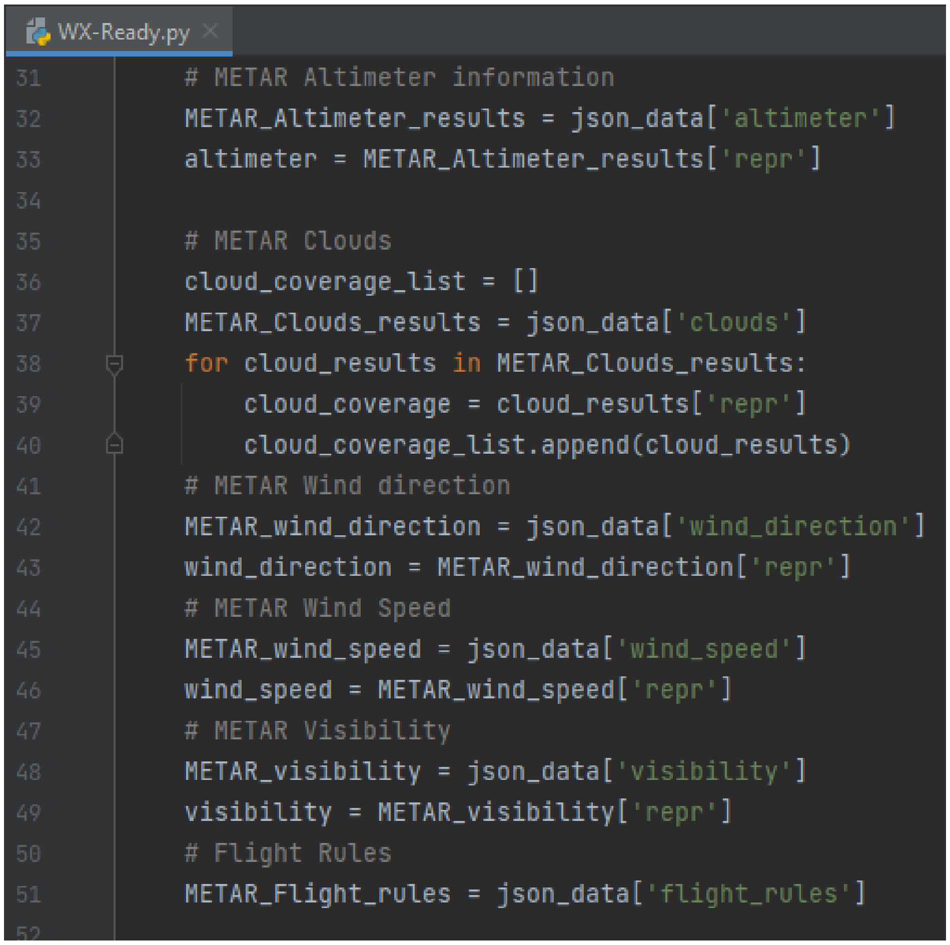
The airport list contains all the airports the crew inputs into the UI. We then authenticate to the endpoint using the bearer token method. After the authentication, we modify the server response to a JSON format.

8.2 Backend METAR Info Retrieval "JSON data consumption."
Now that our response is in JSON format as shown in figure 11, we place each weather observation in its variable, as shown in figure 12.



```
{
    "meta": {
        "timestamp": "2021-07-26T01:41:06.859177Z",
        "stations_updated": "2021-07-10",
        "cache-timestamp": "2021-07-26T01:40:48.254000Z"
    },
    "altimeter": {
        "repr": "A2897",
        "value": 28.97,
        "spoken": "two eight point nine seven"
    },
    "clouds": [
        {
            "repr": "SCT004",
            "type": "SCT",
            "altitude": 4,
```

FIGURE 11: METAR weather data in JSON format.

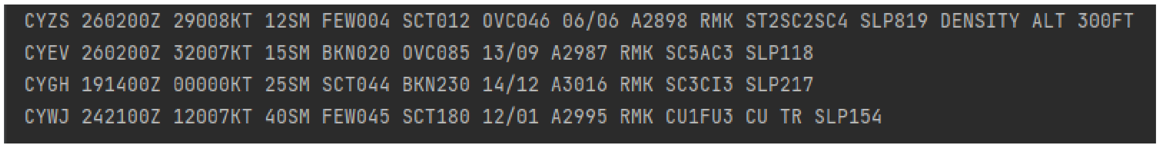


```
    # METAR Altimeter information
    METAR_Altimeter_results = json_data['altimeter']
    altimeter = METAR_Altimeter_results['repr']

    # METAR Clouds
    cloud_coverage_list = []
    METAR_Clouds_results = json_data['clouds']
    for cloud_results in METAR_Clouds_results:
        cloud_coverage = cloud_results['repr']
        cloud_coverage_list.append(cloud_results)
    # METAR Wind direction
    METAR_wind_direction = json_data['wind_direction']
    wind_direction = METAR_wind_direction['repr']
    # METAR Wind Speed
    METAR_wind_speed = json_data['wind_speed']
    wind_speed = METAR_wind_speed['repr']
    # METAR Visibility
    METAR_visibility = json_data['visibility']
    visibility = METAR_visibility['repr']
    # Flight Rules
    METAR_Flight_rules = json_data['flight_rules']
```

FIGURE 12: Python Code to dissect JSON data into variables.

8.3 Backend Weather Info Display In METAR Format

We can display the information gathered to the crew by printing the variables while respecting the International Civil Aviation Organization ICAO METAR format clearly shown in figure 13.

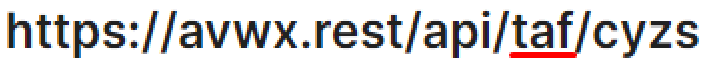


```
CYZS 260200Z 29008KT 12SM FEW004 SCT012 OVC046 06/06 A2898 RMK ST2SC2SC4 SLP819 DENSITY ALT 300FT
CYEV 260200Z 32007KT 15SM BKN020 OVC085 13/09 A2987 RMK SC5AC3 SLP118
CYGH 191400Z 00000KT 25SM SCT044 BKN230 14/12 A3016 RMK SC3CI3 SLP217
CYWJ 242100Z 12007KT 40SM FEW045 SCT180 12/01 A2995 RMK CU1FU3 CU TR SLP154
```

FIGURE 13: METAR weather info displayed in ICAO format.

8.4 Backend TAF Info Retrieval "GET request."

This section of the backend code is the same as the METAR section. The only difference is that the TAF section uses its endpoint, as shown in figure 14.

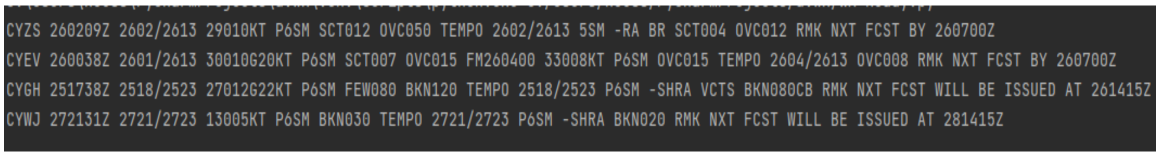


https://avwx.rest/api/taf/cyzs

FIGURE 14: TAF API Endpoint.

8.5 Backend Weather Info Display In TAF Format

In figure 15 We display the TAF information retrieved to the crew by printing the variables while respecting the ICAO TAF format.

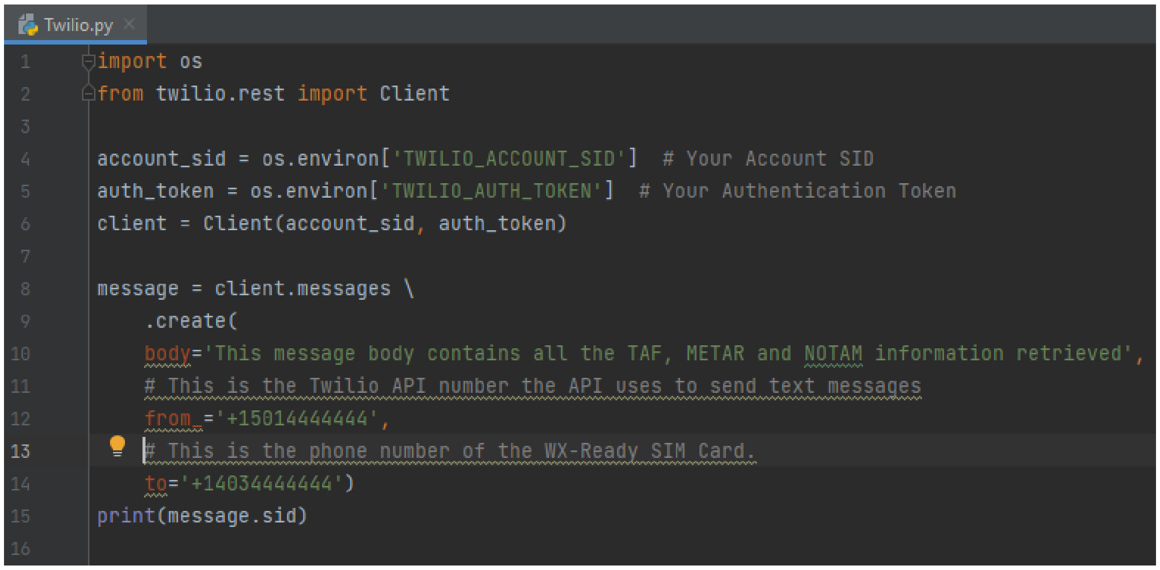


```
CYZS 260209Z 2602/2613 29010KT P6SM SCT012 OVC050 TEMPO 2602/2613 5SM -RA BR SCT004 OVC012 RMK NXT FCST BY 260700Z
CYEV 260038Z 2601/2613 30010G20KT P6SM SCT007 OVC015 FM260400 33008KT P6SM OVC015 TEMPO 2604/2613 OVC008 RMK NXT FCST BY 260700Z
CYGH 251738Z 2518/2523 27012G22KT P6SM FEW080 BKN120 TEMPO 2518/2523 P6SM -SHRA VCTS BKN080CB RMK NXT FCST WILL BE ISSUED AT 261415Z
CYWJ 272131Z 2721/2723 13005KT P6SM BKN030 TEMPO 2721/2723 P6SM -SHRA BKN020 RMK NXT FCST WILL BE ISSUED AT 281415Z
```

FIGURE 15: TAF weather info displayed in ICAO format.

8.6 Backend Sending Weather Data through Cellular Network

Once we have gathered all the weather data, the Twilio messaging service will reply to the same phone number that requested the info shown in figure 16.



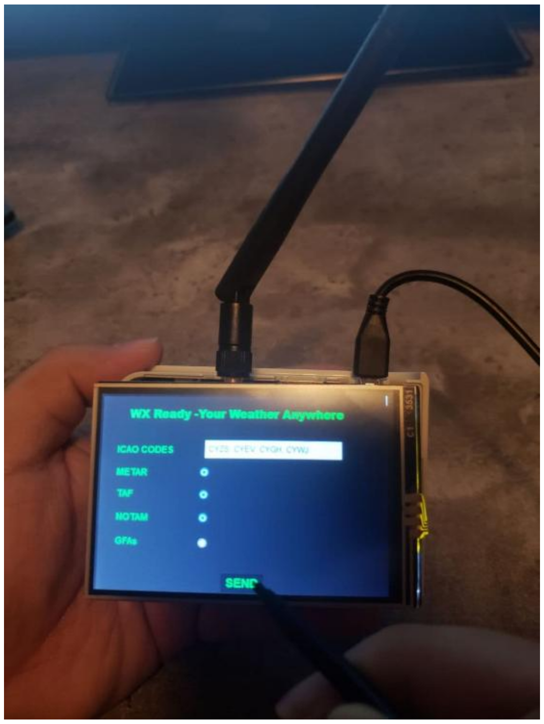
Twilio.py

```python
import os
from twilio.rest import Client

account_sid = os.environ['TWILIO_ACCOUNT_SID']  # Your Account SID
auth_token = os.environ['TWILIO_AUTH_TOKEN']  # Your Authentication Token
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
    body='This message body contains all the TAF, METAR and NOTAM information retrieved',
    # This is the Twilio API number the API uses to send text messages
    from_='+15014444444',
    # This is the phone number of the WX-Ready SIM Card.
    to='+14034444444')
print(message.sid)

```

FIGURE 16: Weather data transmission via SMS using Twilio API.

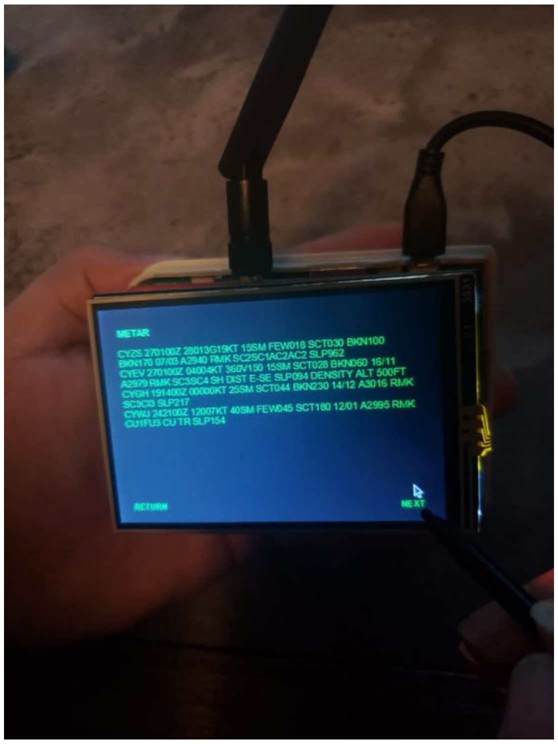
9. THE WORKING PROTOTYPE

This section shows how WX-Ready is operating in figure 17, figure 18, figure 19 and figure 20.



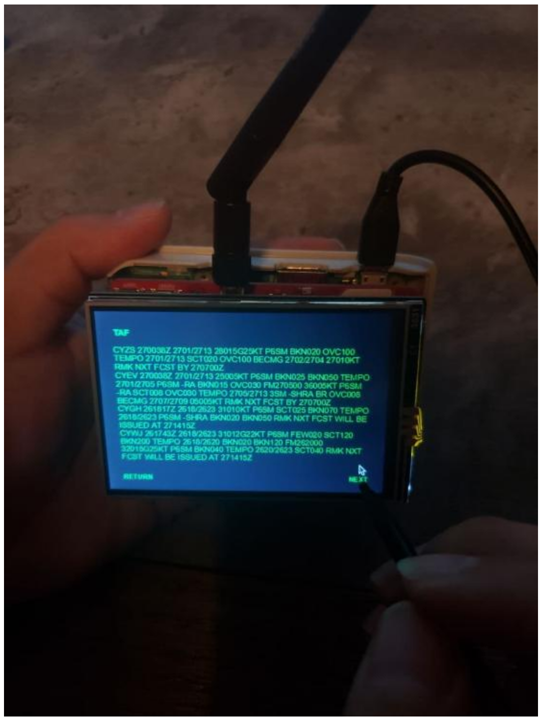
```
CYZS CYEV CYGH CYWJ
```

FIGURE 17: WX-Ready main screen.



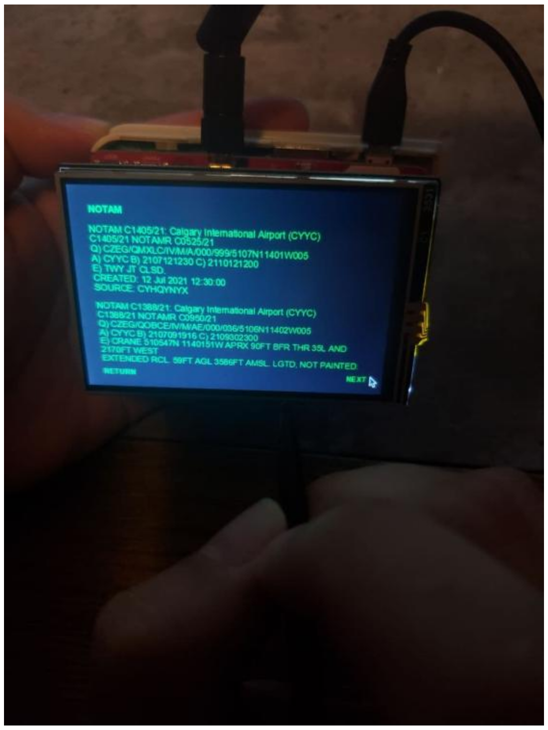
```
CYZS 270100Z 28013G19KT 15SM FEW018 SCT030 BKN100
BKN170 07/03 A2940 RMK SC2SC1AC2AC2 SLP962
CYEV 270100Z 04004KT 360V150 15SM SCT028 BKN060 16/11
A2979 RMK SC3SC4 SH DIST E-SE SLP094 DENSITY ALT 500FT
CYGH 191400Z 00000KT 25SM SCT044 BKN230 14/12 A3016 RMK
SC3C8 SLP217
CYWJ 242100Z 12007KT 40SM FEW045 SCT180 12/01 A2995 RMK
CU1FU3 CU TR SLP154
```

FIGURE 18: WX-Ready METAR screen.



```
TAF

CYZS 270038Z 2701/2713 28015G25KT P6SM BKN020 OVC100
TEMPO 2701/2713 SCT020 OVC100 BECMG 2702/2704 27010KT
RMK NXT FCST BY 270700Z
CYEV 270038Z 2701/2713 25005KT P6SM BKN025 BKN050 TEMPO
2701/2705 P6SM -RA BKN015 OVC030 FM270500 36005KT P6SM
-RA SCT008 OVC030 TEMPO 2705/2713 3SM -SHRA BR OVC005
BECMG 2707/2709 05005KT RMK NXT FCST BY 270700Z
CYGH 261817Z 2618/2623 31010KT P6SM SCT025 BKN070 TEMPO
2618/2623 P6SM -SHRA BKN020 BKN050 RMK NXT FCST WILL BE
ISSUED AT 271415Z
CYWJ 261743Z 2618/2623 31012G22KT P6SM FEW020 SCT120
BKN200 TEMPO 2618/2620 BKN020 BKN120 FM262000
32015G25KT P6SM BKN040 TEMPO 2620/2623 SCT040 RMK NXT
FCST WILL BE ISSUED AT 271415Z
```

FIGURE 19: WX-Ready TAF screen.



```
NOTAM

NOTAM C1405/21: Calgary International Airport (CYYC)
C1405/21 NOTAMR C0525/21
Q) CZEG/QMXLC/IV/M/A/000/999/5107N11401W005
A) CYYC B) 2107121230 C) 2110121200
E) TWY JT CLSD.
CREATED: 12 Jul 2021 12:30:00
SOURCE: CYHQYNYX

NOTAM C1388/21: Calgary International Airport (CYYC)
C1388/21 NOTAMR C0950/21
Q) CZEG/QOBCE/IV/M/AE/000/036/5106N11402W005
A) CYYC B) 2107091916 C) 2109302300
E) CRANE 510547N 1140151W APRX 90FT BFR THR 35L AND
2170FT WEST
EXTENDED RCL. 59FT AGL 3586FT AMSL. LGTD, NOT PAINTED.
```

FIGURE 20: WX-Ready NOTAM screen.

10. FUTURE ENHANCEMENTS

WX-Ready is eventually intended to operate in the most remote areas in the world. To reach this goal, the apparatus needs to send the requests via the GSM network and through the Iridium satellite network when the GSM network is not available. Since it is based on the Raspberry PI hardware platform, the WX-Ready Apparatus can easily be connected to a Rockblock Mk2 Iridium modem shown in figure 21. This modem can send SMS messages through the Iridium Satellite Constellation via the SBD Short Burst Data communication protocol. Adding this capability to WX-Ready does not change the systems' architecture discussed above because all we do is sending the SMS message through a different network. The backend servers and processes remain the same.

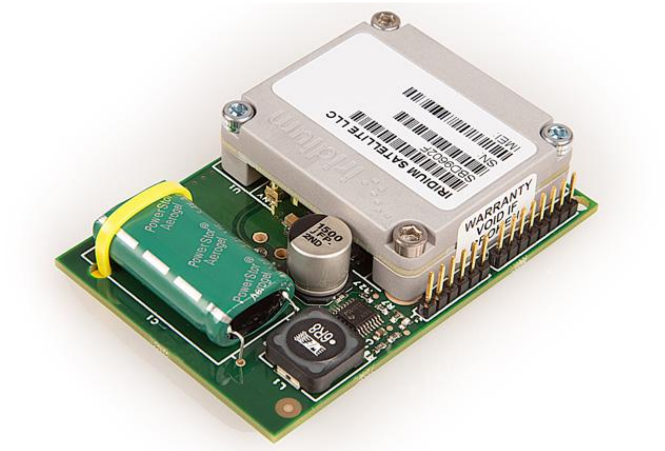


FIGURE 21: RockBlock Satellite Modem.

11. CONCLUSION

The developed prototype performed its job correctly. In the future, we should add additional features to the software and upgrades to the hardware. Currently, the system can operate anywhere in Canada, provided it's within Cellular coverage. To have WX-Ready work virtually anywhere globally, we can add an Iridium transceiver that can communicate with the Iridium satellite constellation using the Short Burst Data protocol (SBD). Currently, we are getting the power from a USB connection; however, we can also add a battery to provide power to the device. WX-Ready costs less than 40$ and has an operating cost of 10$ per month. This apparatus is the cheapest way to retrieve aviation weather information in remote areas versus commercially available ACARS solutions with prices ranging from $200.000 to $400.000.

WX-Ready benefits airline operators in the Arctic and remote regions like Africa, South America, and Asia.

12. REFERENCES

Anaman, Kwabena & Quaye, Ruth & Owusu-Brown, Bernice. (2017). Benefits of Aviation Weather Services: A Review of International Literature. *Research in World Economy.* 8. 45-58. 10.5430/rwe.v8n1p45.

Aviation Weather Services (2009). Advisory Circular 00-45F, Change 2. *Federal Aviation Administration.*

Government of Canada, O. of the A. G. of C.(2017, May 16). *Report 6—Civil Aviation Infrastructure in the North—Transport Canada*. Www.oag-Bvg.gc.ca. https://www.oag-bvg.gc.ca/internet/English/parl_oag_201705_06_e_42228.html

International Civil Aviation Organization (ICAO).(2010, July).*Annex 3 to the Convention on International Civil Aviation: Meteorological Service for International Air Navigation, 20th ed.*

Schwartz, B., & Benjamin, S. G. (1995). A Comparison of Temperature and Wind Measurements from ACARS-Equipped Aircraft and Rawinsondes. *Weather and Forecasting, 10*(3), 528–544. https://doi.org/10.1175/1520-0434

Smith, M., Strohmeier, M., Lenders, V., & Martinovic, I. (2016). On the security and privacy of ACARS. *Integrated Communications Navigation and Surveillance (ICNS).* Published. https://doi.org/10.1109/icnsurv.2016.7486395

Susanto, A., & Meiryani.(2019).*System Development Method with The Prototype Method.* International Journal of Scientific and Technology Research8(07) pp. 142–143

Tamalet, S., Gobbo, G., Durand, F., & Deville, J. G. (2007). Acars router for remote avionics applications (US8484384B2).*United States Patent.* https://patents.google.com/patent/US8484384B2/en