Spectrum Alerting System Based on Software Defined Radio and Raspberry Pi

David Ball, Nitin Naik and Paul Jenkins

Defence School of Communications and Information Systems

Ministry of Defence, United Kingdom

Outline

- Motivation
- Proposed Design of Spectrum Alerting System (SAS) based on RTL-SDR and Raspberry Pi
- Experimental Results of the proposed SAS
- Conclusion and Future Work

Motivation

- Traditionally the analyst would manually tune a radio receiver across the frequency band to identify unusual or ‘rogue’ signals.
- This was very time consuming and expensive in terms of resources.
- With the advent of digital radio, the ‘sweep’ of the frequency band could be programmed into the radio receiver which automatically highlighted frequencies of interest.
- If a difference is detected, according to pre-set thresholds, an alert is generated and analysed accordingly.

Motivation..

- However, signals intelligence operations can be inhibited by

1. High cost digital radio receivers and processors.
2. Paucity of expensively trained Signals Intelligence Analysts (SIA).

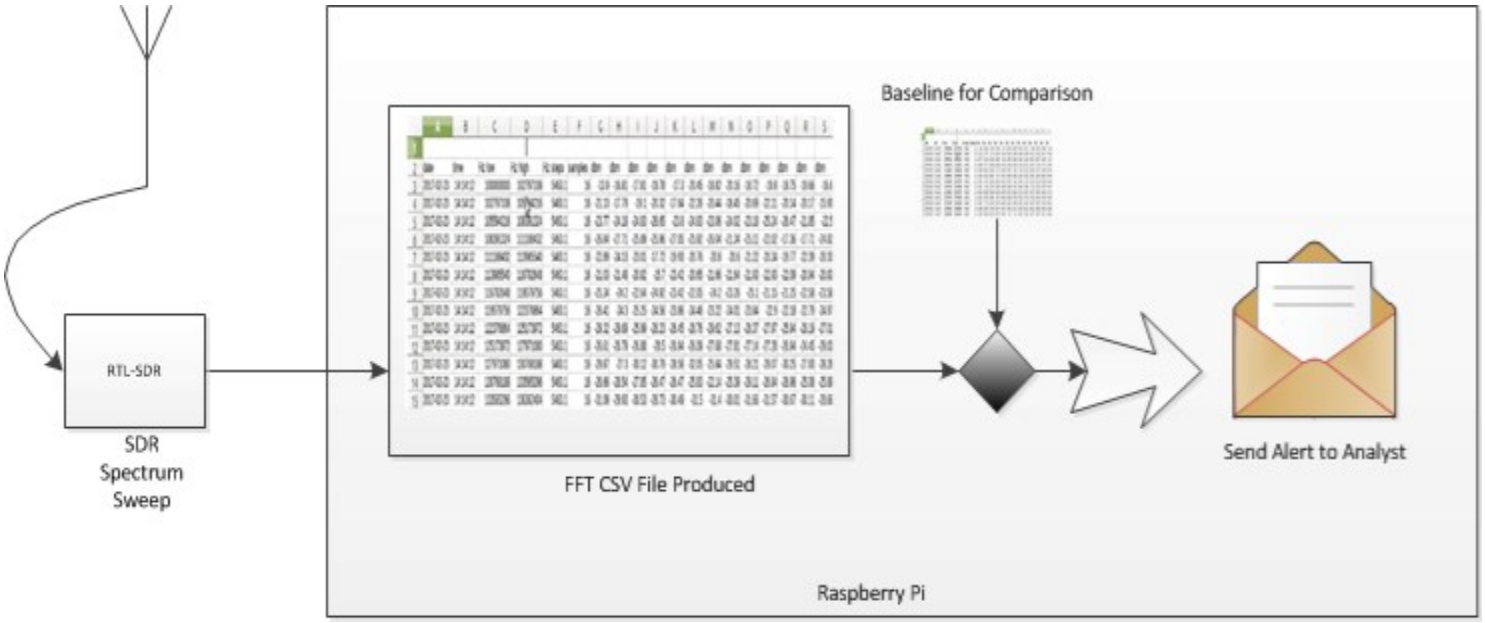
- The use of lightweight, cost-effective, low-cost SDR and low-power microprocessors can significantly reduce the cost.
- Several low-cost SDR devices are available in the market and RTL-SDR USB device is one of the cheapest SDR.
- Similarly, Raspberry Pi is one of the low-cost and low-power microprocessor devices available.

Motivation..

- Automatically generating and sending an alert message to Signals Intelligence Analysts (SIA) can save both time and money.
- SIA may decide to use the information to re-baseline the system, or they may decide that the frequency is of interest and take further intelligence action.
- Therefore, this paper presents the design and implementation of an inexpensive and generic Spectrum Alerting System (SAS), based on RTL-SDR USB device and a Raspberry Pi that can send an alert to SIA.

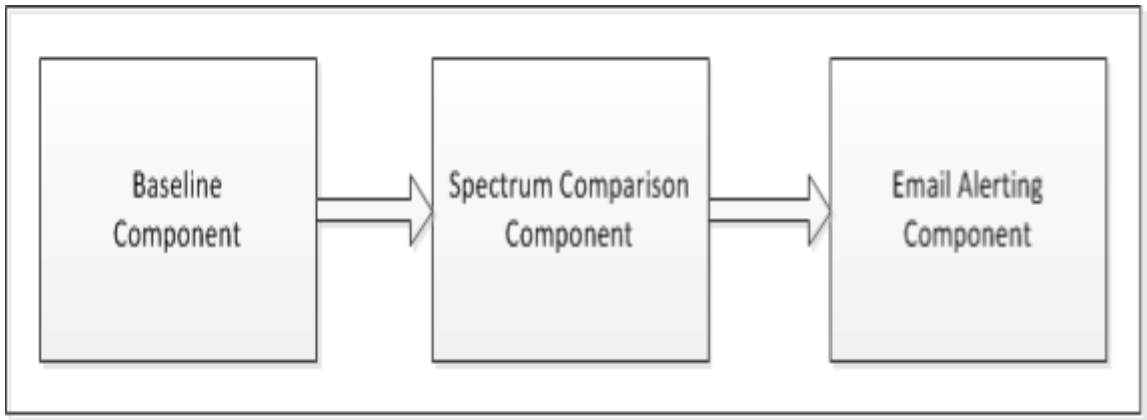
Proposed Design of SAS

- Figure shows the proposed design model of Spectrum Alerting System (SAS) and its operational flow.
- In which, RTL-SDR USB device, Python and Raspberry Pi are utilised.
- RTL-SDR USB device acquires signals, presenting them to the Raspberry Pi. The signal is sampled and passed to a Fast Fourier Transform block.
- The software module of the SAS performs the baselining, signal comparison and alerting via email.



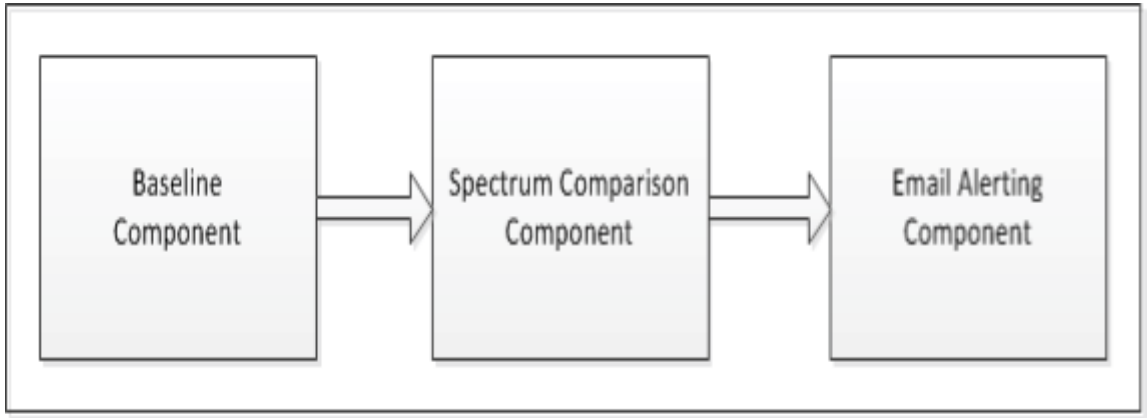
Proposed Design of SAS (1)

- The software module of the SAS is written in Python and developed in three stages: baseline component, spectrum comparison component and email alerting component.
- The first stage is a baseline component to capture a sample of the EM spectrum and record it in the CSV file format.
- The design of this baseline component requires following operations:
- Sets the frequency range, i.e., lower limit and upper limit (in MHz).
- Sets the FFT bin size (in Hz).
- Sets the gain (in dB).
- Sets the length of the capture (in seconds).
- Assigns the output filename for the CSV file.



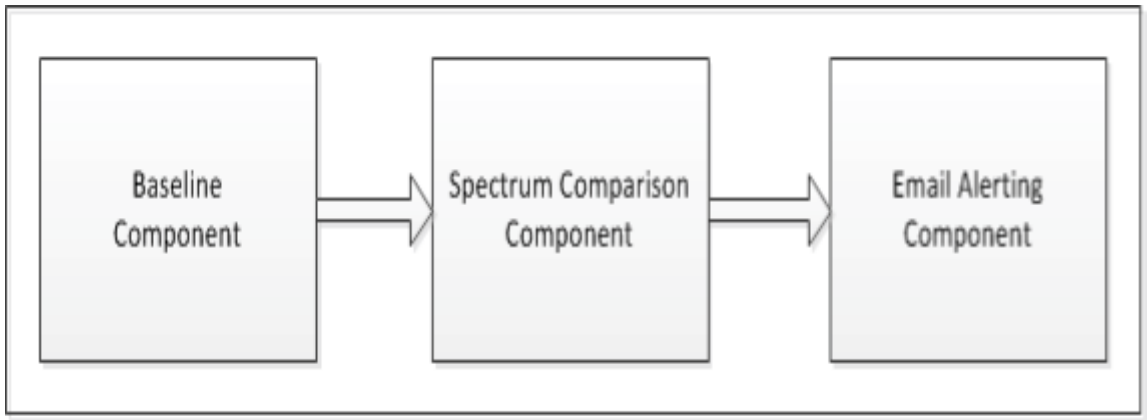
Proposed Design of SAS (2)

- The second stage of the SAS software module is to continuously sample the spectrum of interest, at specified intervals recording the readings, comparing the reading with the baseline.
- Both baseline and the new reading are stored on the system as CSV files.
- It subtracts the baseline from the new reading and records the result; any differences between the two, are subsequently compared with the threshold, if the threshold is breached, the program automatically proceeds to issue an email alert.
- If the threshold is not breached, the program returns to its sampling mode, collecting another sample reading and the whole process repeats.



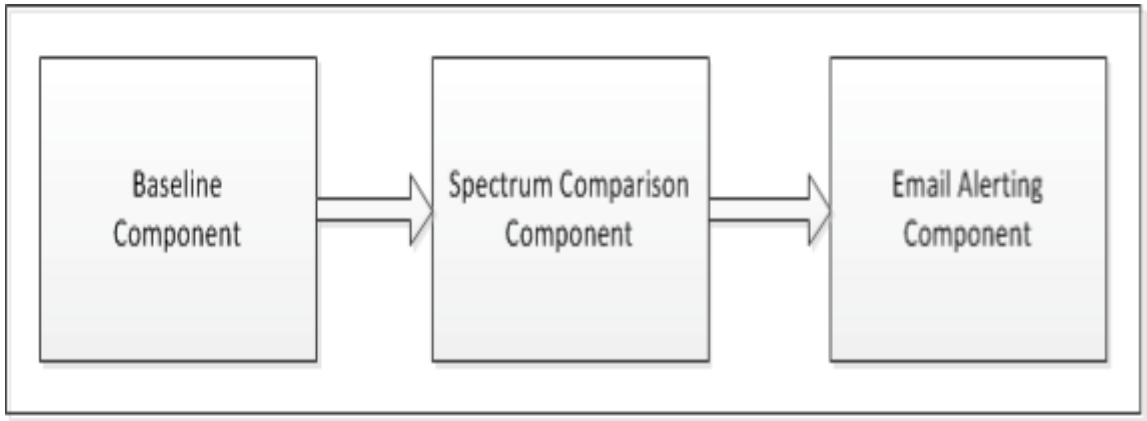
Proposed Design of SAS (3)

- The design of this spectrum comparison component requires following operations:
- Use the same spectrum parameters as the baseline.
- Assign the output filename for the CSV file (on the initial comparison capture).
- Capture the spectrum data and append the comparison CSV file.
- Subtract the Baseline from the comparison spectrum.
- Test the result against the thresholds.
- Step on to stage three to send email alert if the threshold is breached, otherwise return for the next comparison capture.



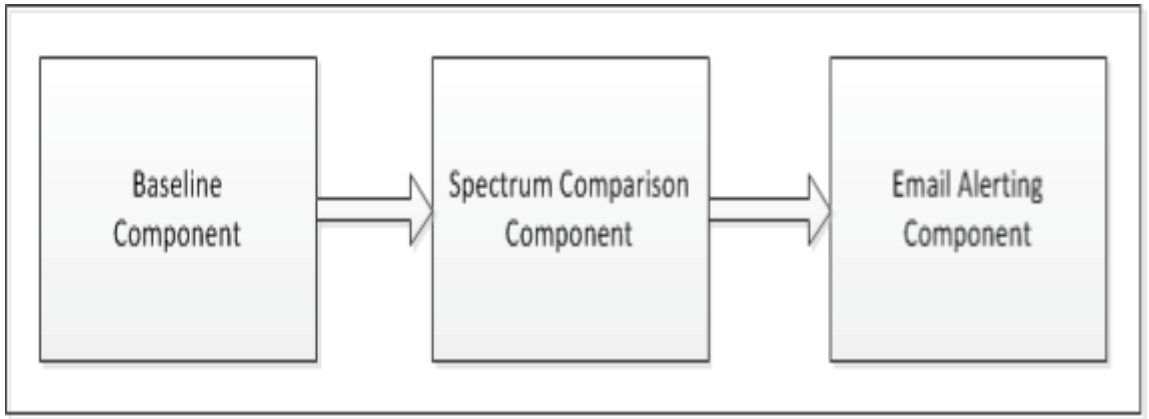
Proposed Design of SAS (4)

- The final third stage of the software module of the SAS is an email alerting component to trigger an email alert on an occurrence of a specified event.
- The conditions were identified which would prevent the remote signals intelligence analyst from being bombarded with emails.
- Firstly, an email would only be sent if a specified event occurred; the specified event being the breach of the threshold value in the comparison results.
- Secondly, an email would only be sent if 300 seconds have elapsed since the last email.
- This would prevent the continuous sending of emails in the event of the first condition being continually met.
- These conditions are set for this particular implementation, however, these conditions can be easily adjusted by the user depending on their requirements.



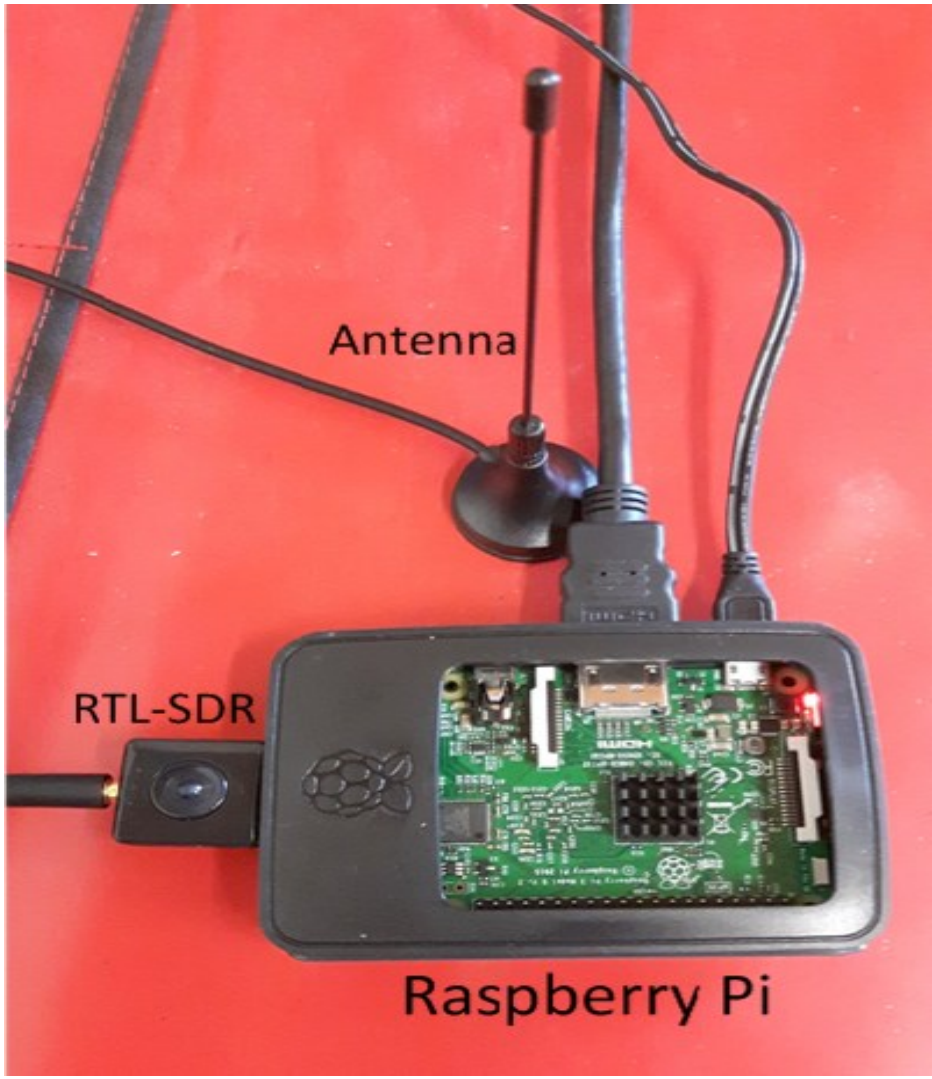
Proposed Design of SAS (5)

- The design of the email alerting component requires following operations:
- Importing the Simple Message Transfer Protocol (SMTP) and Email Python modules.
- Configuring the desired email format and specifying the email account, subject and body along with any documents to be attached.
- Configuring the SMTP server including login details of the internet-based email account.
- Instructing the email program to send an email if the event is True.
- Updating ‘time since last transmission’ CSV file with ‘time now’, if email is sent.



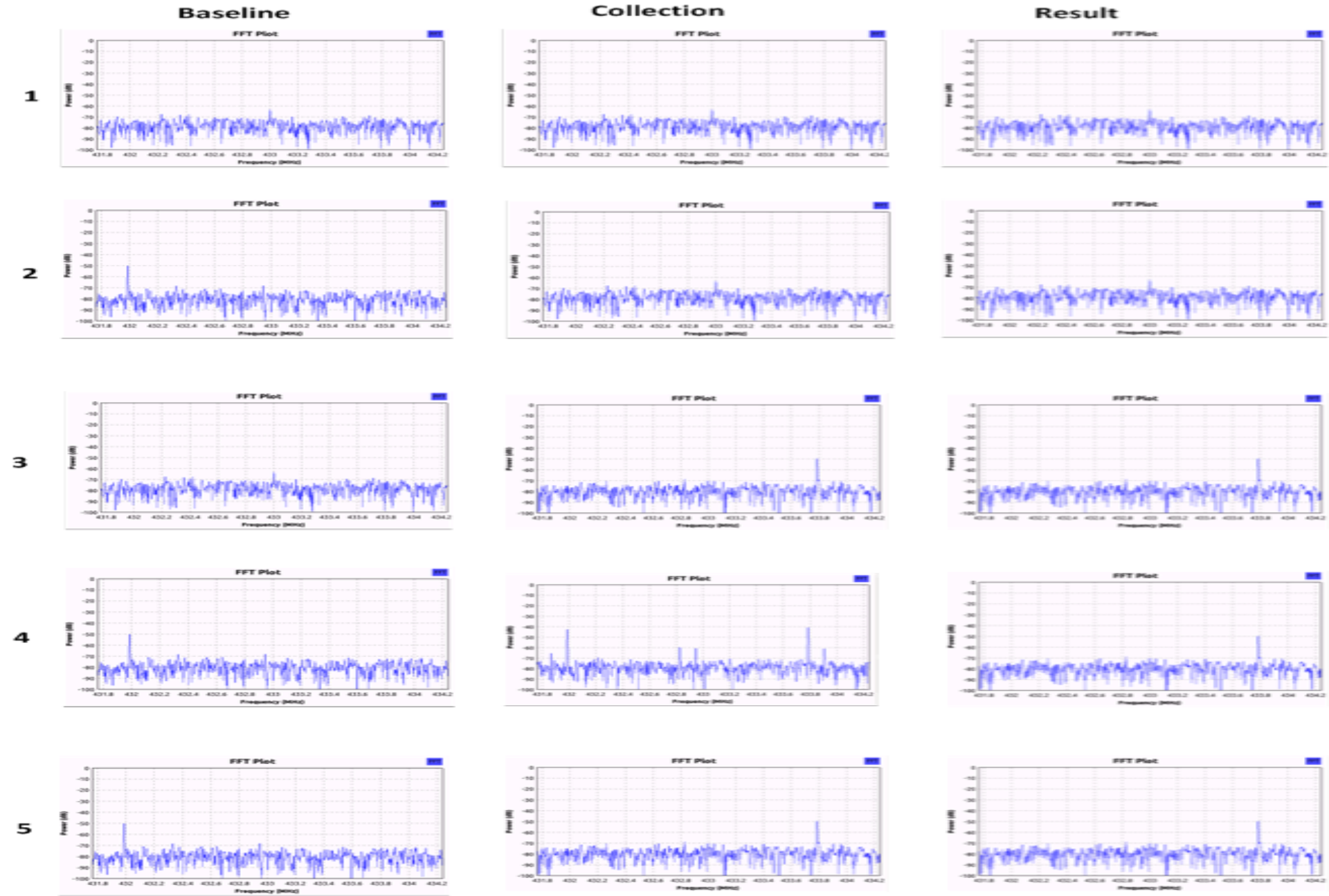
Proposed Design of SAS (6)

- The output CSV file was designed to be appended to, rather than written over.
- This would allow a history of new signals to be recorded within the output file.
- However, to produce the graph to attach to the email, only the most recent line in the CSV file would be required.



SAS Built using RTL-SDR and Raspberry Pi (ARM Cortex)

Experimental Results of the proposed SAS



Conclusion and Future Work

- This paper has presented the design and successful implementation of an inexpensive and generic Spectrum Alerting System (SAS) based on Software Defined Radio (SDR) by using the RTL-SDR USB device and a low-power microprocessor Raspberry Pi.
- The system captured an instance of the baseline spectrum and stored it in the CSV file for comparison.
- On subsequent spectrum sweeps the code compared the newly capture spectrum CSV file with the baseline CSV file.
- If the threshold is breached, the SAS generates an email alert of the spectrum differences to a remote signals intelligence analyst.
- The signals intelligence analyst has the option to remotely access the SAS to reset threshold or re-baseline the spectrum captured depending on their analysis.

Conclusion and Future Work..

- The implementation and experimental results demonstrate the success of the proposed SAS in the particular scenario.
- It shows the potential of SAS as an inexpensive and generic choice for an intelligence surveillance system in both a benign and hostile environment.
- Although the RTL- SDR works well for scanning a small frequency range for this particular experiment, if more complex waveforms are to be successfully decoded then a more complex SDR receiver is required.
- In the future, it is important to examine this spectrum alerting system for other real-time applications and compare its results with the established radio system to determine its real success and suitability for specific projects.

Thank You
&
Questions ?

Email: BallD@DSCIS.ac.uk