

Uma arquitetura para radar secundário nos modos A, C e S com processamento todo em software

Vitor Augusto Ferreira Santa Rita
Centro Tecnológico do Exército, Rio de Janeiro, RJ, Brasil

Resumo — Neste artigo é apresentada uma arquitetura para um radar secundário operando nos modos A, C e S que habilita o processamento a ser feito todo em software. Nas arquiteturas tradicionais, há um estágio inicial de processamento em hardware, num chip FPGA, com posterior processamento em linguagem de baixo nível. Nesta arquitetura proposta, o estágio inicial comumente feito em hardware é substituído por um processo, em software, de SO em alta prioridade implementado em baixo nível. E os estágios posteriores substituídos por uma implementação em alto nível, reduzindo a complexidade do projeto, melhorando manutenibilidade e reduzindo o tempo de codificação. Para validar esta arquitetura, foi implementado um simulador com condições reais de simulação executado numa placa de desenvolvimento ARM com reduzida capacidade de processamento. O simulador ainda teve o estágio de recepção validado com dados reais.

Palavras-Chave — Radar secundário, padrão ICAO modo S, arquitetura processamento

I. INTRODUÇÃO

Os modos de operação A e C para radares secundários, com o passar dos tempo, estão se tornando cada vez mais obsoletos para controle de tráfego aéreo. São modos nos quais as perguntas e respostas são feitas em *broadcast* e causam a saturação do canal de comunicação.

Em ambientes densos em rotas de voo ocorre a sobreposição de respostas, com conseqüente impossibilidade de decodificar as informações, causando falta de informação correta aos controladores de tráfego aéreo. Esse problema foi evidenciado em acidentes ocorridos na década de 1960, nos Estados Unidos, onde foi identificado que a causa de tais acidentes foi a sobreposição de respostas dos modos A e C.

O padrão modo S surgiu como uma atualização dos modos A e C, tendo sido projetado há 40 anos pelo *Massachusetts Institute of Technology*, oferecendo arquiteturalmente a minimização dos problemas comuns de FRUIT e GARBLING [1].

Apesar do padrão modo S existir por mais de 40 anos ele não é largamente adotado mundialmente em parte pela sua maior complexidade de implementação comparado aos modos A e C.

Entretanto, a necessidade do padrão modo S para o controle de tráfego aéreo é evidente hoje. Nele é possível obter um número maior de detalhes sobre o voo de uma aeronave por uma comunicação direcionada pelo número ICAO (international civil aviation organization) da aeronave.

E parte da especificação do modo S, o ADS-B (*Automatic Dependent Surveillance Broadcast*) é particularmente interessante comercialmente pois permite a comunicação de informação de voo tendo a aeronave que ser equipada apenas com uma *transponder* transmissor omnidirecional, e o radar de base apenas com receptor omnidirecional.

O ADS-B tem mostrado sua aplicabilidade e versatilidade em prover informações de voo, tendo sido implementado até em *transponders* para drones, por exemplo os da família XP da Sagetech (<http://www.sagetechcorp.com>).

Este trabalho apresenta uma arquitetura de software com reduzida complexidade para o processamento de sinais de um radar secundário operando nos modos A, C e S, permitindo que o projeto de um radar secundário possa ser levado à implementação – incluindo o padrão modo S – em menor custo, tempo e complexidade.

A arquitetura implementada para transmissão e recepção nos modos A, C e S teve o estágio de recepção validado com dados reais com dados coletados a partir de um receptor de RF.

II. VISÃO GERAL

As capacidades propostas por esta arquitetura estão resumidas a seguir.

- Distância: 100 km
- Precisão em azimute: $\sim 1^\circ$
- Precisão em distância: 15 m
- Capacidade de processamento: ≥ 140 aeronaves
- Período de rotação da antena: 4 segundos

Condições de ambiente

O primeiro aspecto considerado para habilitar o processamento ser feito todo em software está relacionado à taxa de amostragem quanto à granularidade da posição de uma aeronave no mapa.

Considerando que o tamanho de uma aeronave é dezenas de metros e que o mapa no qual a aeronave será posicionada tem largura de 200 km (raio de 100 km), é suficiente haver uma precisão em distância de poucos metros para boa granularidade de visualização. Assim, é adequado definir a taxa de amostragem em 10 MHz que dá uma precisão em distância de 15 m.

Identificação dos bits lógicos do padrão modo S

A taxa de amostragem sendo definida em 10 MHz também é suficiente para detectar todos os bits lógicos da especificação dos modos A, C e S: os bits lógicos dos modos A e C são todos posicionados como múltiplos de 0,05 μ s relativo ao primeiro pulso (com duração cada de 0,45 μ s mais ou menos 0,1 μ s); o a posição dos bits do padrão modo S são todos posicionados como múltiplos de 0,5 μ s do primeiro pulso. Sendo assim, a posição de todos pulsos é múltipla de $1/10\text{MHz} = 0,1 \mu\text{s}$. Os bits de posição do padrão A e C estão apresentados na Fig. 1.

Pulsos	Posições (microsegundos)
C1	1,45
A1	2,90
C2	4,35
A2	5,80
C4	7,25
A4	8,70
X	10,15
B1	11,60
D1	13,05
B2	14,50
D2	15,95
B4	17,40
D4	18,85

Fig. 1: Posição dos bits lógicos de informação dos modos A e C (valores extraídos do anexo 10 à Convenção da Aviação Civil Internacional).

Com base nessas características, foi implementado um módulo de detecção de preâmbulo em linguagem de baixo nível com capacidade de processamento de 100 MB/s quando executado num processador Intel Core i7, e de 20MB/s quando executado numa placa de desenvolvimento com processador ARM 1GHz e 1GB de memória RAM, provando a viabilidade de se implementar uma arquitetura para processamento de toda a lógica dos modos A, C e S. Apesar dessa possibilidade, o simulador teve de ser implementado com o objetivo de provar que existe a possibilidade de interpretar as informações anexas aos preâmbulos (*payloads* do modo S) em linguagem de alto nível em um processador de propósito geral. O custo de processamento dos modos A e C é ínfimo quando comparado ao modo S e não será tratado nas seções a seguir.

Descrição do simulador

No simulador foram implementados tanto o radar de estação base quanto o *transponder* de aeronave.

A arquitetura do software do radar de estação base está apresentada na Fig. 2.

Os dados recebidos pelo receptor de RF – SDR USRP – são continuamente filtrados pelo módulo de detecção de preâmbulo com cada preâmbulo identificado tendo o correspondente *payload* marcado com um *timestamp*.

Cada par *payload+timestamp* é enviado para processamento posterior nos módulos de alto nível no processamento do modo S. O módulo de detecção de preâmbulo, em software, reduz a taxa de dados de 20 MB/s para alguns KB/s pois permite a passagem apenas da

informação útil ao remover o ruído do sinal. Cada *payload* repassado para processamento posterior tem 56 ou 112 bits de informação, sendo cada bit representado na recepção em fase e em quadratura com dois bytes de dados. Assim, cada *payload* tem no máximo $112*2 = 224$ bytes de dados.

Considerando a quantidade de aeronaves monitoradas no entorno do radar como sendo da ordem de 100 unidades, há $100*28*224$ bytes sendo recebidos a cada 4 segundos, tomando por base a recepção de 28 respostas por passagem do feixe da antena de transmissão pelo alvo. Isso resulta em 153KB/s. Ou seja, uma taxa útil da ordem de KB/s.

O *timestamp* enviado permite o cálculo da distância da aeronave ao radar. As simulações mostraram que o *timestamp* marcado em software após a recepção de dados, em função das oscilações nos tempos de execução do software, provocaram oscilação nos valores de distância, ocasionando uma imprecisão no posicionamento das aeronaves. Este resultado mostrou que, para uma melhora na precisão do cálculo de distância, apenas a marcação dos dados recebidos deve ser feita em hardware, ou que o processo de marcação de *timestamp* seja executado em um sistema operacional de tempo real.

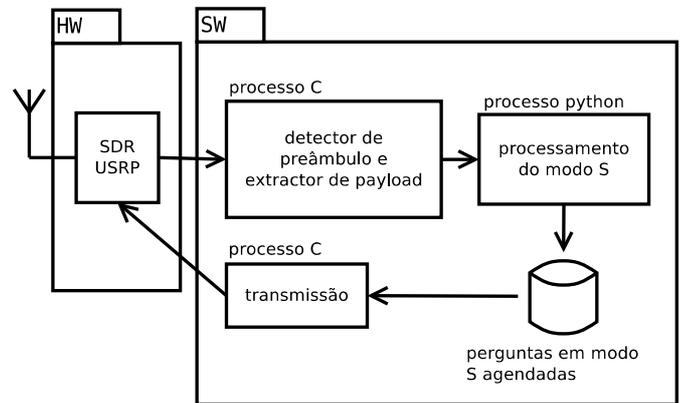


Fig. 2: Diagrama em blocos da arquitetura.

Os módulos de processamento dos *payloads* dos modos A, C e S são diferentes.

Os modos A e C requerem processamento simples: a distância da aeronave é calculada pelo *timestamp* do *payload*, o azimute é calculado como uma média ponderada das múltiplas recepções durante o tempo de exposição da aeronave à iluminação da antena, sendo o peso da média ponderada a amplitude do sinal, e o código de resposta decodificado de acordo com a especificação dos padrões dos modos A e C.

O módulo de processamento do padrão modo S é mais complexo pois requer agendamento de perguntas, com as perguntas em modo seletivo devendo ser enviadas na passagem seguinte de iluminação da antena sobre o alvo.

Similarmente, o modo S calcula a distância da aeronave pelo *timestamp* do *payload*, o azimute é calculado como uma média ponderada das múltiplas recepções durante o tempo de exposição da aeronave à iluminação da antena, e a informação do modo S recebida é armazenada numa base de dados de aeronaves que guarda todas as informações enviadas do *transponder*.

O agendamento de perguntas é feito notificando o módulo de transmissão com a posição futura em azimute da aeronave por meio de uma previsão linear do azimute.

O módulo de transmissão, atua sobre uma matriz na qual cada linha é uma pergunta a ser transmitida no momento em que o azimute da posição de apontamento da antena corresponde à linha. O módulo de agendamento preenche as linhas da matriz de acordo com a previsão de posição das aeronaves com questões do tipo *roll-call* e *all-call* de acordo com a requisição de se obter informações das aeronaves. O número de linhas da matriz define a precisão em azimute desta arquitetura que são 2048 linhas, resultando em uma precisão de $0,17^\circ$.

O simulador tem 1200 linhas de código em linguagem C, 3000 linhas de código em linguagem Python e 200 linhas em linguagem Go.

A arquitetura do *transponder* de aeronave é similar ao do radar de estação base com a simplificação de que a resposta a uma pergunta é instantaneamente enviada, e não há a necessidade de agendamento.

A arquitetura implementada provê manutenibilidade, facilidade na depuração e velocidade no desenvolvimento comparado a implementação com uma etapa em hardware.

Para melhorar a realidade das simulações, um módulo de ruído foi implementado para gerar continuamente ruído aleatório para ser mistura às perguntas e resposta dos padrões dos modos A, C e S.

Para simular a rotação da antena, o simulador gera um valor de azimute a partir do relógio do sistema operacional.

As posições de latitude e longitude são decodificadas do formato CPR (*compact position report*).

A implementação do simulador tomou por base a documentação dos padrões internacionais do modo S [2,3,4,5,6].

Avaliação sobre a capacidade de processamento da arquitetura

Cada resposta modo S demora 1 ms para ser processada de acordo com as simulações o que resulta na possibilidade de processamento de 4000 aeronaves durante a revolução da antena.

Considerando que cada aeronave gera muitas respostas durante a iluminação da antena numa largura de feixe de 5° (valor estimado) com 2048 perguntas por 4 segundos, cada aeronave gera 28 respostas por iluminação. Isso implica em $4000/28 \sim 140$ aeronaves tratáveis em tempo real nesta arquitetura.

Pontos chaves na implementação da arquitetura

Detector de preâmbulo e extrator de payload

O módulo detector de preâmbulo recebe os dados do receptor de RF em blocos, armazenando-os em um array.

A seguir são calculados os valores absolutos de cada amostra IQ *inplace*: cada amostra IQ ($\text{int8}+\text{int8}$) é convertida no valor absoluto (int16). Para aumentar o desempenho, uma *lookup table* de valores absolutos foi codificada e é

computada a cada instanciação do módulo de detecção de preâmbulo.

O próximo passo na busca de preâmbulos é encontrar o menor dos valores altos de bits, chamado aqui “A”, e o maior dos valores baixos de bits, chamado aqui “B”. Uma vez esses calculados, se o valor absoluto da diferença for maior que um *threshold* o ponto de checagem é dito ter um preâmbulo.

O absoluto da diferença de A e B quando feita ao final da checagem do preâmbulo confere uma taxa de tratamento de dados que é melhorada em 300% quando a checagem é feita a cada amostra avaliada do preâmbulo.

Consolidação do valor de azimute

A cada vez que uma aeronave é iluminada pela antena os valor de azimute recebidos são consolidados com uma média ponderada em função da amplitude para resultar no valor do azimute da iluminação. A recepção de azimutes ocorre desde a recepção do primeiro azimute por até 2 segundos após, para se ter a certeza de que todas as respostas de uma aeronave foram recebidas. Uma vez o tempo seja transcorrido para uma aeronave um valor único de azimute é consolidado e atribuído a instância da aeronave. No cálculo do azimute, uma média ponderada é calculada baseada na amplitude do sinal recebido.

Com o valor consolidado, este é inserido numa fila de azimutes da aeronave que permite a estimação por extrapolação da posição em azimute futura da aeronave, para o agendamento de pergunta exclusiva.

Lógica de alto nível implementada em Python

A implementação da lógica de processamento do modo S foi implementada em linguagem Python e conferiu velocidade na implementação da especificação, facilidade de depuração com o uso de estruturas de dados complexas não existentes em linguagem de baixo nível.

Além disso, Python diminuiu o tempo de projeto por permitir que, uma vez o simulador implementado, o código fosse diretamente executado na plataforma de destino sem necessidade de adaptação.

III. COMPATIBILIDADE E ADEQUAÇÃO DA IMPLEMENTAÇÃO AO PADRÃO MODO S

Software

A emulação do *transponder* não atende à especificação quanto ao tempo de resposta dos modos A, C e S que deve ser menor que $128 \mu\text{s}$ para o modo S, e menor que $3 \mu\text{s}$ para os modos 3A e C. Vide a seguir os itens relativos do anexo 10 da ICAO.

3.1.1.7.10 Reply delay and jitter. The time delay between the arrival, at the transponder receiver, of the leading edge of P3 and the transmission of the leading edge of the first pulse of the reply shall be 3 plus or minus 0.5 microseconds. (...)

3.1.2.10.3.8.2 Reply delay and jitter for Mode S. For all input signal levels between MTL and -21 dBm, the leading edge of the first preamble pulse of the reply (3.1.2.2.5.1.1) shall occur 128 plus or minus 0.25 microsecond after the sync phase reversal (3.1.2.1.5.2.2) of the received P6. (...)

O tempo de resposta do *transponder* é maior que o tempo requerido pela especificação em virtude do processamento em software. Essa característica obrigou o radar de base a ser implementado levando em consideração o tempo de resposta do *transponder* emulado, diferente da especificação, a fim de que as distâncias das aeronaves emuladas fossem corretamente calculadas. Entretanto, essa realidade emulada não impede que, no caso real, o *reply delay* seja configurado no radar de base com o tempo previsto no anexo 10 da ICAO. Assim, a arquitetura proposta é adequada para o radar de base, mas não para a construção de *transponder*.

De acordo com a especificação, a estação base executa operações de pergunta de *all-call* e *roll-call* por meio dos pacotes de subida UF4, UF5, UF11, UF20 e UF21.

Analogamente, o *transponder* é capaz de responder a cada uma dessas perguntas por meio dos pacotes de descida DF0, DF4, DF5, DF11, DF16, DF17, DF20, DF21, DF24. E também transmite continuamente pacotes do tipo DF17 (ADS-B) para reportar automaticamente a posição GPS.

Foram implementados também os recursos de *lockout*, checagem de paridade e codificação e decodificação no formato CPR.

IV. RESULTADOS EXPERIMENTAIS

O simulador foi testado em duas configurações: uma com recepção real e uma com transmissão e recepção emulada.

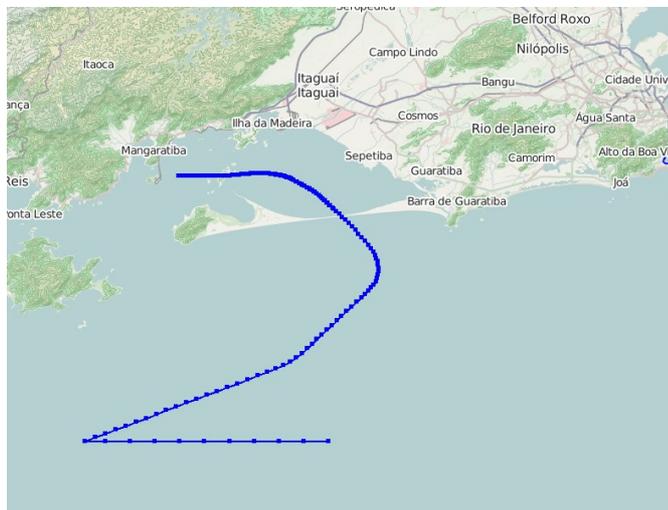


Fig. 3: Trajetórias reais de duas aeronaves acompanhadas pelo radar de estação base por meio dos pacotes de descida DF17 enviados automaticamente pelos *transponders* das aeronaves.

Na recepção real, com um exemplo de acompanhamento de aeronave na Fig. 3, os pacotes do tipo DF17 foram enviados pela aeronave e corretamente interpretadas pelo radar de base executado numa placa de desenvolvimento ARM, validando a cadeia de recepção e capacidade da arquitetura de processamento.

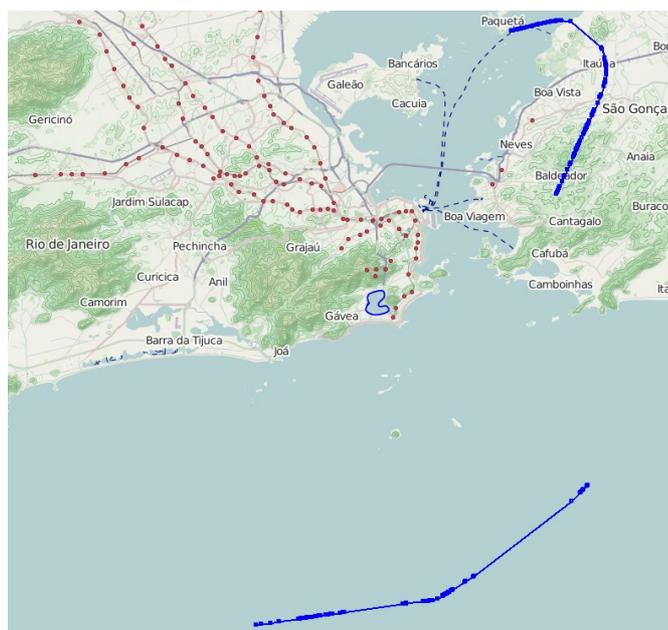


Fig. 4: Aeronave com trajetória emulada acompanhada pelo radar de base.

Na outra configuração de teste, na Fig. 4, uma aeronave emulada e controlada via linha de comando, teve sua trajetória corretamente acompanhada pelo radar de base. Neste caso, os *handshakes* de *all-call* e *roll-call* foram corretamente executados, e a posição da aeronave extrapolada para pergunta da aeronave na volta seguinte da antena foi executada com sucesso, inclusive numa manobra abrupto emulada. Com esse resultado, pode-se mostrar uma robustez no algoritmo de rastreamento.

V. CONCLUSÃO

Neste trabalho foi apresentada uma arquitetura de radar secundário para processamento de sinais todo em software, incluindo o padrão modo S.

Esta arquitetura visa substituir um estágio inicial de processamento que normalmente é feito em hardware (por um chip FPGA) por um processo de alta prioridade em software implementado em linguagem C a fim de que o projeto de um radar secundário modo S possa ser executado com menor complexidade, menor tempo, menor custo e com maior facilidade de implementação, depuração e com maior modularidade.

VI. OBSERVAÇÕES FINAIS

Este trabalho foi desenvolvido baseado unicamente na documentação internacional do padrão modo S.

REFERÊNCIAS

- [1] V. A. Orlando, "The mode S beacon radar systems", The Lincoln Laboratory Journal, vol. 2, nº 3, p. 345-362, 1989.
- [2] International Civil Aviation Organization, "Aeronautical telecommunications", Annex 10, vol. IV, July 2007.
- [3] International Civil Aviation Organization, "Technical provisions for mode s services and extended squitter", doc 9871, an/464, 2008.

- [4] International Civil Aviation Organization, “Manual on mode s specific services”, Doc 9688, AN/952, Second Edition, 2004.
- [5] P. Hunt, “CRC calculation for mode-s transponders”, Eurocontrol Experimental Centre, European Organization for the Safety of Air Navigation Eurocontrol, December 1994.
- [6] P. Hunt, L. Crouzard, “Aircraft position report using dgps and mode-s”, Eurocontrol Experimental Centre, February 1995.