

SISTEMA DE BÚSQUEDA DE NAÚFRAGOS DF 935 (DIRECTION FINDER)

Jose Alberto Benítez Andrades

@jabenitezdev

jabenitez@gmail.com

ÍNDICE

- 0. Introducción
- 1. Estructura del simulador
 - 1.1. PI – HOST – CV – CREAL
 - 1.2. Desarrollo de módulos
- 2. Chelton DF935
 - 2.1. Análisis
 - 2.2. Desarrollo y pruebas realizadas
- 3. Demostración en cabina virtual
- 4. Vídeos y fotos del simulador
- 5. Conclusiones
 - 5.1. Posibles mejoras
 - 5.2. Conclusiones finales

INTRODUCCIÓN

- Lenguaje: GNU C++
- Dirigido por: Jorge Balbás Rodrigo
- Supervisado por: Miguel Ángel Benítez Andrades
- Presencia necesaria en el CES, durante el desarrollo y pruebas
- Detalle:
 - Estudio de la metodología de desarrollo de módulos de simulación de Indra. Programación orientada a objetos bajo entorno Linux.
 - Estudio del equipo real a simular DF935.
 - Desarrollo de interface HMI gráfico simulando el interface real e integración en una consola simulada.
 - Simulación del equipo de detección de naufragos (DF935).
 - Integración de dicho sistema en el simulador de Helicóptero EC225 en el desarrollo con Indra para EC UK.

INTRODUCCIÓN

- CHELTON DF 935



INTRODUCCIÓN

- Helicóptero EC225



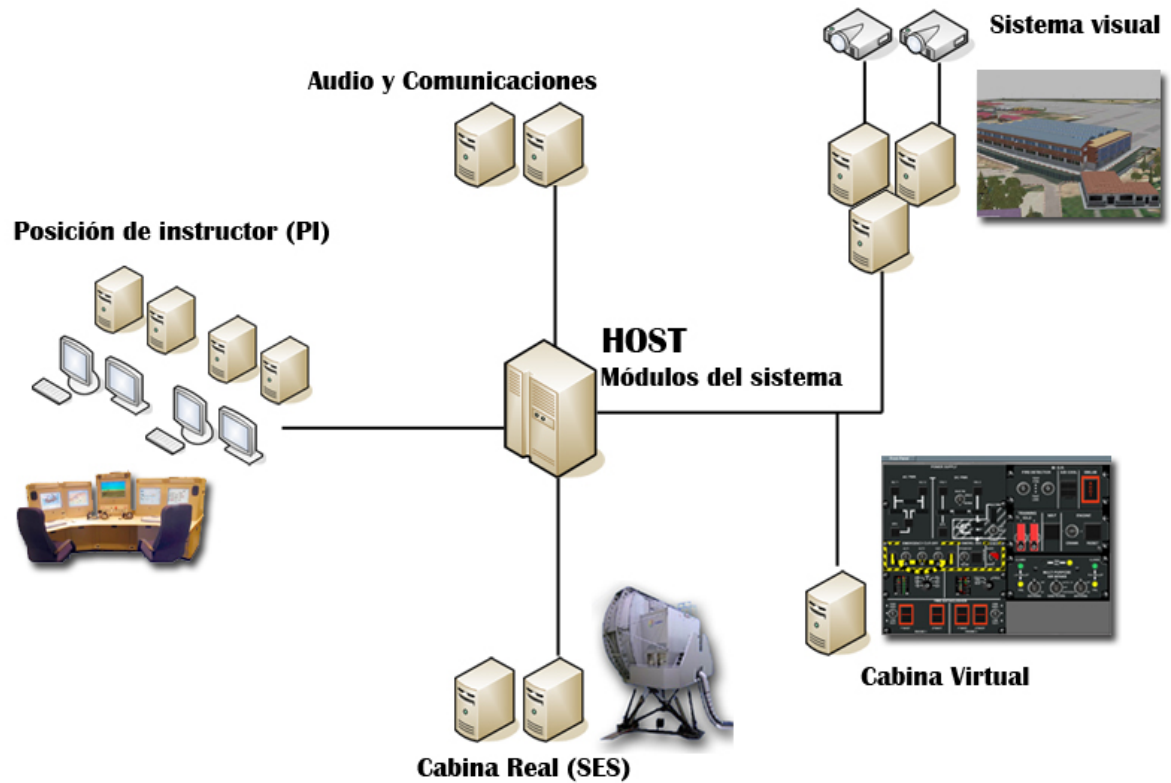
INTRODUCCIÓN

- Cabina del helicóptero EC225



ESTRUCTURA DEL SIMULADOR

- PI – HOST – C.VIRTUAL – C.REAL



ESTRUCTURA DEL SIMULADOR

- Desarrollo de módulos

Desarrollo de módulos



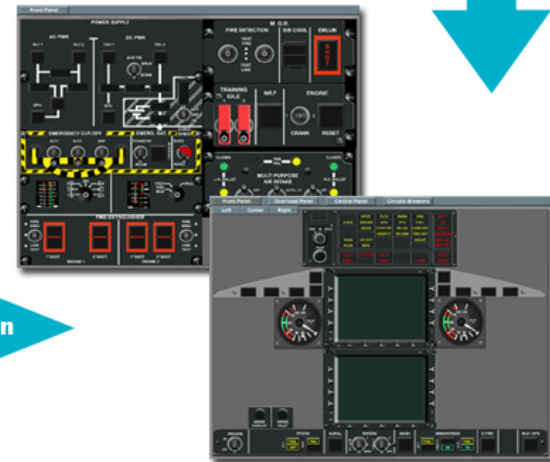
Pruebas

Tests



Integración

Cabina virtual



Diseño de paneles

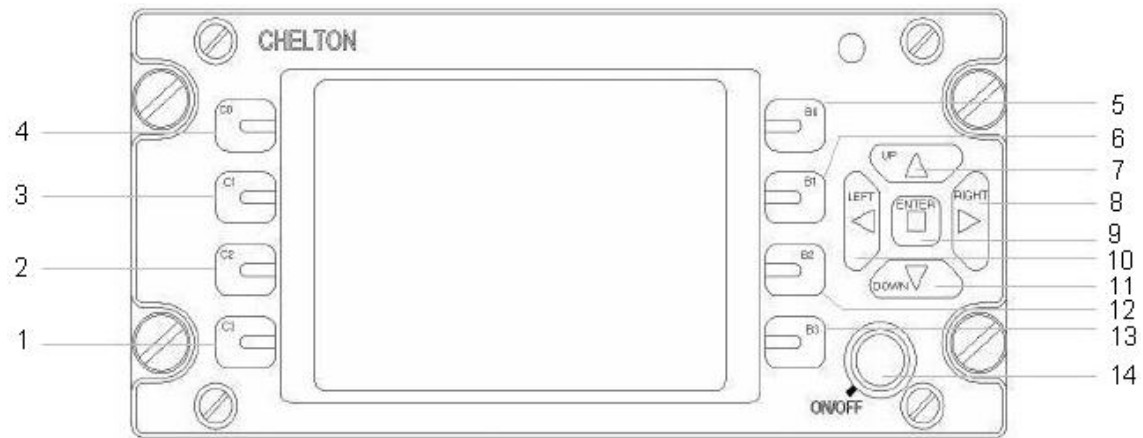


Integración

CHELTON DF 935

1. Análisis

- Estudio de la documentación
 - Funcionamiento del dispositivo
 - Estructura del dispositivo





CHELTON DF 935

1. Análisis

- Resolución de dudas
- Elección de metodología de trabajo
 - Creación de pantallas en glStudio
 - Desarrollo de la parte lógica del dispositivo en su correspondiente módulo.

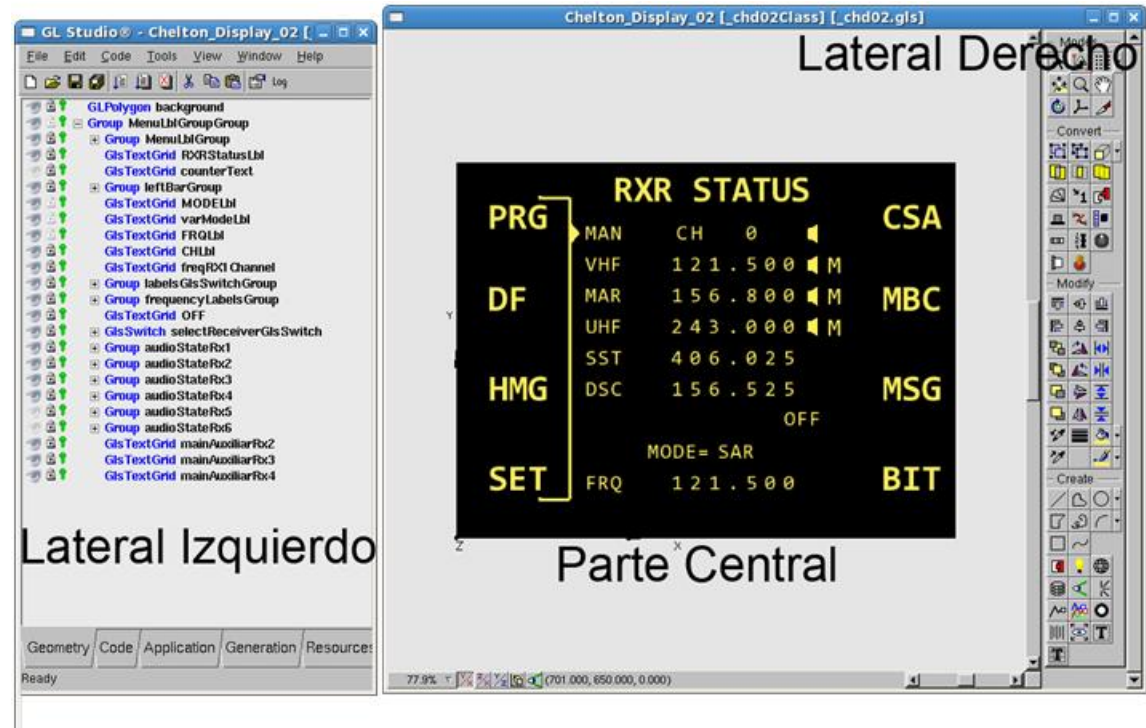
CHELTON DF 935

2. Desarrollo y pruebas realizadas

- Creación de la parte gráfica
 - Aprendizaje de la herramienta glStudio
 - Programación en GNU C++
 - Prueba de los distintos menús
- Creación del módulo y pruebas
 - Creación del módulo que contiene la parte lógica del dispositivo
 - Realización de pruebas en los distintos cálculos necesarios
- Unión entre el módulo y la parte gráfica

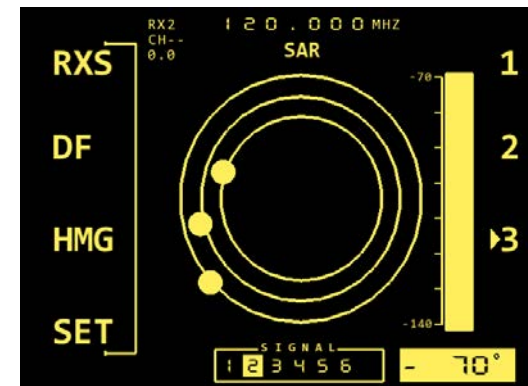
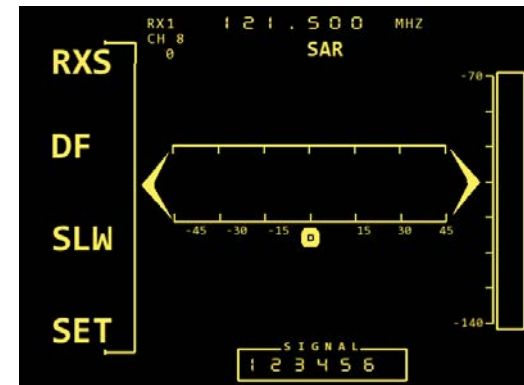
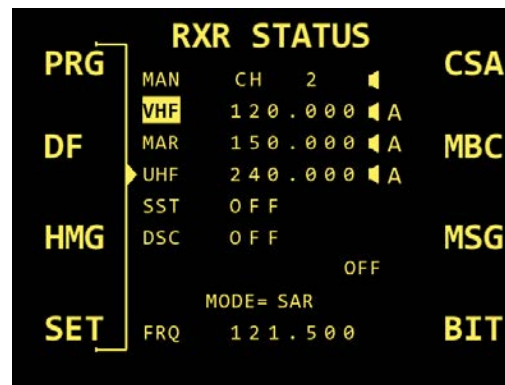
CHELTON DF 935

- glStudio



CHELTON DF 935

- Pantallas del Chelton DF 935 [1]



CHELTON DF 935

- Pantallas del Chelton DF 935 [2]

```

PROG RX
RXS  RX4      UHF
      CHANNEL --
DF    FREQ 240.000
      ON/OFF  ON
      SQ.TYP  NOISE
      LEVEL   -112
HMG  ▶AUDIO  NORMAL
      B/W     WIDE
      MOD     AUTO
SET

```

```

SETUP DF
RXS  VOLUME 00 ◻
DF    BOTTOM MNT
HMG  UHF MBR
SET  DSP   Ver  1.000
      BF   Ver  23.45

```

```

CONTROLLER
RXS  FLASH ON-TOP      Y
DF    BEARING AGE      N
      BRIGHTNESS      3 ◻
HMG  AUTO MULTI        N
SET  INHIBIT TEST      N
      35507-E V1.02

```

```

DF REPORT
RXS  SYNTH: OK
      PSU  : OK
DF    I2C  : OK
      A/D  : OK
      TEMP: OK
HMG  ANTEN: OK
      COMMS: OK
SET

```

CHELTON DF 935

- Creación del módulo
 - Ficheros fuente a crear:
 - Hmcheltondf.h /.cpp
 - HmcheltondfBase.h /.cpp
 - HmcheltondfSystem.h /.cpp
 - Hmcheltondf_parameters.h
 - Hmcheltondf_struct.h
 - Hmcheltondf_structlocal.h
 - Métodos destacados en System:
 - ModuleExecution
 - CalculateBeaconData
 - GetCover
 - GetGeodesicDistance

CHELTON DF 935

■ CalculateBeaconData [1]

```
void cI_CheltondfSystem::CalculateBeaconData(int rx, float frequencySelected)
{
    // For 0 to 4 ( We receive data of 5 beacon [4 boats & 1 Cast away])
    for (int index = 0; index < MAX_NUM_BEACONS; index++)
    {
        chelton_out->beacon[index].state = input->beaconSar[index].state;

        // Beacon not detected (by default)!!
        chelton_out->beaconDetected[rx][index] = false;

        // Beacon ACTIVE
        if (chelton_out->beacon[index].state)
        {
            // Set data about a beacon
            m_PositionBeacon.SetLatLon(input->beaconSar[index].latitude, input->beaconSar[index].longitude);

            // Compare the beacon frequency with frequency selected
            if (input->beaconSar[index].frequency == frequencySelected)
            {
                // Calculate the geodesic distance relative to Hel
                chelton_out->beacon[index].distance = GetGeodesicDistance(input->Hel.dLatitude,
                    input->Hel.dLongitude,
                    input->beaconSar[index].latitude,
                    input->beaconSar[index].longitude);

                // If the geodesic distance is < coverRadius ...
                if (chelton_out->beacon[index].distance < coverRadius)
                {

```


CHELTON DF 935

■ CalculateBeaconData [2]

```
// Beacon detected!!
chelton_out->beaconDetected[rx][index] = true;

// Calculate the bearing relative, age and the strength (inverse to distance)
chelton_out->beacon[index].bearingRel = m_Receptor.RadialRelative(input->beaconSar[index].bearingRel - input->beaconSar[index].bearingRel + 360 + 180, 360) - 180;
chelton_out->beacon[index].age = 0;
chelton_out->receptorDetectedBeacons[rx] = true;
chelton_out->beacon[index].strength = 1/chelton_out->beacon[index].distance;
} // end if beacon.distance < coverRadius

} // end_if (input->beaconSar[index].frequency == frequencySelected)

} // end_if (chelton_out->beacon[index].state)
else
{
// Si no está activa ...
chelton_out->beacon[index].strength = 1;
countdTime += dTime;
if (countdTime > 10)
{
countdTime = 0;
chelton_out->beacon[index].age++;
}
}

// Sort the beacons by strength
SortBeacons();

} // End_for (int index = 0; index < MAX_NUM_BEACONS; index++)
}
```

CHELTON DF 935

■ GetCover

```
double cI_CheltondfSystem::GetCover()
{
    double height = input->Hel.dAltitude;
    double cover = sqrt(height + ( 2 * height * EARTH_RADIUS * 1000));

    // Get the cover of DF
    if (cover > MIN_DETECTION_DISTANCE)
        return cover;
    else
        return MIN_DETECTION_DISTANCE;
} // End of GetCover
```

■ GetGeodesicDistance

```
double cI_CheltondfSystem::GetGeodesicDistance (double lat1, double long1, double lat2, double long2)
{
    double dlong;
    double dvalue;
    double dd;

    dlong = (long2 - long1);

    dvalue = (sin(lat1 * DEGTORAD) * sin(lat2 * DEGTORAD)) + (cos(lat1 * DEGTORAD) * cos(lat2 * DEGTORAD)
    * cos(dlong * DEGTORAD));
    dd = acos(dvalue) * RADTODEG;

    return(dd * 111302);
}
```

DEMOSTRACIÓN EN CABINA VIRTUAL

The screenshot displays a virtual cockpit environment with several windows and instrument displays:

- RXR STATUS**: A central status window showing parameters for PRG, DF, HMG, SET, CSA, MBC, MSG, and BIT. The PRG section includes MAN, CH, and 0. The DF section includes VHF (1 2 1 . 5 0 0), MAR (1 5 6 . 8 0 0), and UHF (2 4 3 . 0 0 0). The HMG section includes SST (OFF) and DSC (OFF). The SET section includes MODE= SAR and FRQ (1 2 1 . 5 0 0). The right side shows CSA, MBC, MSG, and BIT.
- Terminal Window**: A terminal window titled "leon:~/ec225/simulador/interfaces" showing a list of files and a command prompt. The files listed are ifc_ec_kin, ifc_ec_mfd, ifcgpws, ifcftn, ifcls, ifcmov, ifcpil35, ifcscs, ifcscs.tgz, ifc_vibration, ifcvis, and ifcvru3, ifcvru4, interface.mk, and test.mk. The terminal also shows a command to run a simulation: "sb -sl 5000 -bg 'black' -fg 'white' -fn 'lucidas -e /home/A350/pruebas/ogmartinez/executive/scr...".
- Instrument Displays**: Several instrument displays are visible, including a primary flight display (PFD) with altitude, airspeed, and heading indicators, and a vertical speed indicator (VSI). The PFD shows a blue sky background with a yellow horizon line and a red airspeed indicator. The VSI shows a scale from 0 to 4000 feet per minute.
- Navigation Display**: A navigation display (ND) showing a heading scale from 0 to 360 degrees and a distance scale from 0 to 30 nautical miles. The heading scale is marked with N, E, S, and W. The distance scale is marked with 0, 10, 20, and 30. The ND also shows a heading error indicator and a distance to go indicator.
- System Windows**: A system window titled "edora" is visible in the bottom right corner, showing the text "edora" in a large font. Another window titled "mfd2" shows a heading scale and a distance scale. A window titled "mfd4" shows a heading scale and a distance scale.



VÍDEOS Y FOTOS DEL SIMULADOR

- Ver vídeos y fotos...

CONCLUSIONES

- Posibles mejoras
 - Sistema de mensajería SST y DSC
 - Botón de chequeo del dispositivo
 - Mayor número de balizas detectadas
- Conclusiones finales
 - Funcionamiento de una empresa grande
 - Conocimiento de protocolos
 - Importancia de una buena organización
 - Relación con el resto del equipo



AGRADECIMIENTOS

- Gracias a todos por asistir a esta presentación