**Librerías faltantes:**

|  |  |  |
| --- | --- | --- |
| **Librería** | **Línea** | **Uso** |
| HVS.h | 3456, 3513, 4542 4824 | Notamos que esta librería se usa en condicionales para el control de temperatura, desconocemos que tipo de estructura es y de donde provienen esos datos.  // Calculate Difference to maxvalue (unit 0,1K)  if (HVS.stateT1E != HVS\_T1E\_IS\_ON)  // offset (island)  if (HVS.stateL1E != HVS\_L1E\_IS\_ON)    // stop condition for mixture temperature  if (HVS.stateT1E != HVS\_T1E\_IS\_ON) |
| ENG.h |  | Necesitamos la estructura de ENG(según feedback enviado corresponde a los datos del motor) para poder mapear variables y tener un entendimiento con el motor que vayamos a implementar. |

**Variables o funciones faltantes:**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Línea** | **Uso** |
| TODO\_OLD | 3392 3495 | #ifdef TODO\_OLD |
| MAIN | 2507 | if (MAIN.Simulation) |
| MAIN\_CONTROL | N/A | No tenemos MAIN\_CONTROL , suponemos que es la función principal que realiza la lógica de cambio de modo de la maquina y demas. |
| CLIENT\_VERSION | 4013 | #if (CLIENT\_VERSION == IET) |
| io\_calculate\_AI\_R\_U\_value  io\_calculate\_AI\_I\_value | 4514 | io\_calculate\_AI\_R\_U\_value( RECEIVER\_TEMP, MIX.ReceiverTemperature.Raw,  PARA[ParRefInd[MIX\_OPTION\_RECEIV\_TEMP\_SENSOR\_\_PARREFIND]].Value, STOPCONDITION\_70226, STOPCONDITION\_70227 ); |
| MIX\_P\_T\_FACTOR\_A\_\_PARREFIND | 3371 | ParIndex = MIX\_P\_T\_FACTOR\_A\_\_PARREFIND |
| MIX\_P\_T\_FACTOR\_B\_\_PARREFIND | 3372 | ParIndex = MIX\_P\_T\_FACTOR\_B\_\_PARREFIND |
| NominalSpeed | 4219 | / (TUR.NominalSpeed / 100) |
| GovernorAnalogOutputInternal | 4221 | **TUR**.GovernorAnalogOutputInternal |
| ThrottlePositionPercent | 4243 | DK.ThrottlePositionPercent |
| AI\_R\_U\_FUNCT | 4511 | AI\_R\_U\_FUNCT[RECEIVER\_TEMP].Assigned |

**Observaciones:**

1. En el archivo MIX.h nos indica una línea de código que la función MIX\_control() debe agregarse a la lista de tareas, pero esa función necesita argumentos de entrada y es llamada por la función Transit(máquina de estados).

//MIX.h

\* @ void MIX\_control() is called from 10Hz control Task

-------------------------------------------------------

//MIX.c

static void MIX\_Control(const DU8 sig, DU8 mixer)