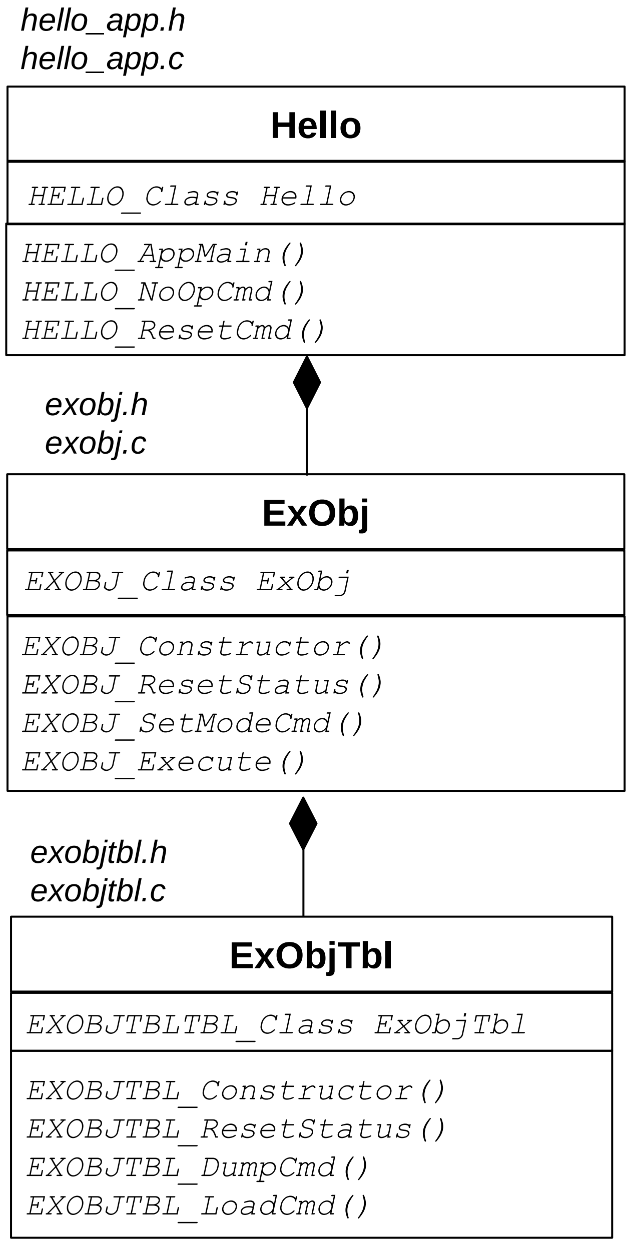
Lesson Objectives

- **Introduce OpenSatKit's application framework design**

- **Modify the 'Hello World' app to learn how to work with the design**
  - Add event messages to example object
  - Change default table values

This an abbreviated tutorial for the Beta release and will be expanded in the future

Application C Framework Overview



hello_app.h
hello_app.c

Hello

HELLO_Class Hello

HELLO_AppMain()
HELLO_NoOpCmd()
HELLO_ResetCmd()

exobj.h
exobj.c

ExObj

EXOBJ_Class ExObj

EXOBJ_Constructor()
EXOBJ_ResetStatus()
EXOBJ_SetModeCmd()
EXOBJ_Execute()

exobjtbl.h
exobjtbl.c

ExObjTbl

EXOBJTBLTBL_Class ExObjTbl

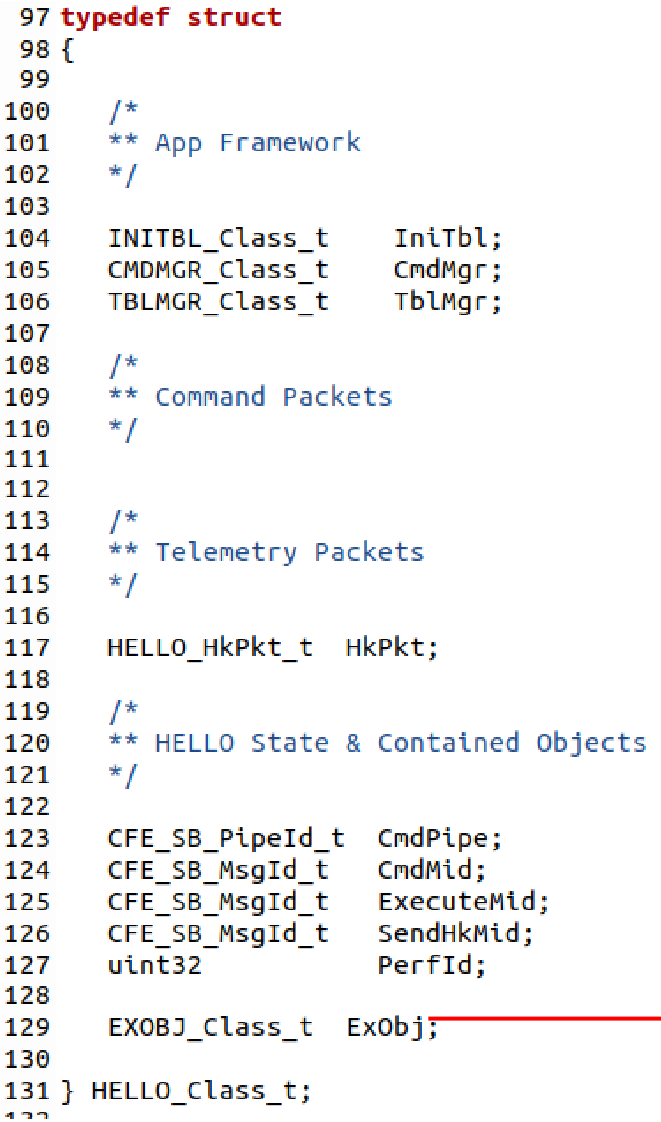
EXOBJTBL_Constructor()
EXOBJTBL_ResetStatus()
EXOBJTBL_DumpCmd()
EXOBJTBL_LoadCmd()

- **The App C Framework is an object-based design written in C**
- **Apps are constructed as an aggregation of objects**
  - Hello contains one Example Object (ExObj)
  - ExObj contains one Example Object Table (ExObjTbl)
  - The object hierarchy can be as wide or deep as needed
- **The key roles of the main app are to**
  - Read the app’s JSON initialization configuration file
  - Initialize contained objects and register their commands
  - Manage the main control loop
- **Contained objects implement the ‘business logic’**
  - ExObj increments a counter during each execution cycle
  - ExObj’s Set Mode command supports increment and decrement
  - ExObjTbl defines the counter’s lower and upper limits

Application Composition

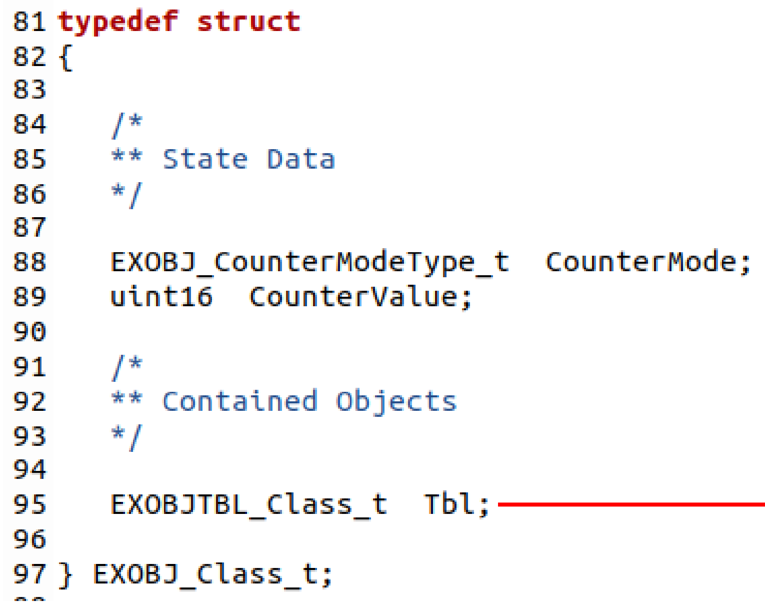
- **Use a variation of the ‘singleton” design pattern**
  - Object constructors passed reference to owner’s storage
  - `void EXOBJ_Constructor(EXOBJ_Class_t* ExObjPtr, const INITBL_Class_t* IniTbl);`
  - Contained objects store reference a static variable so subsequent function (or method) calls don’t require a pointer to be passed

hello_app.h



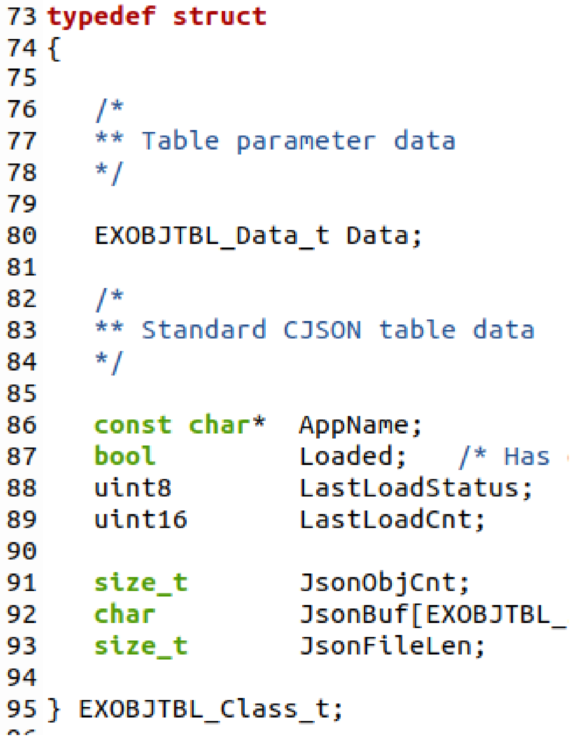
```c
typedef struct
{

   /*
   ** App Framework
   */

   INITBL_Class_t    IniTbl;
   CMDMGR_Class_t    CmdMgr;
   TBLMGR_Class_t    TblMgr;

   /*
   ** Command Packets
   */


   /*
   ** Telemetry Packets
   */

   HELLO_HkPkt_t  HkPkt;

   /*
   ** HELLO State & Contained Objects
   */

   CFE_SB_PipeId_t  CmdPipe;
   CFE_SB_MsgId_t   CmdMid;
   CFE_SB_MsgId_t   ExecuteMid;
   CFE_SB_MsgId_t   SendHkMid;
   uint32           PerfId;

   EXOBJ_Class_t  ExObj;

} HELLO_Class_t;
```

exobj.h



```c
typedef struct
{

   /*
   ** State Data
   */

   EXOBJ_CounterModeType_t  CounterMode;
   uint16  CounterValue;

   /*
   ** Contained Objects
   */

   EXOBJTBL_Class_t  Tbl;

} EXOBJ_Class_t;
```

exobjtbl.h

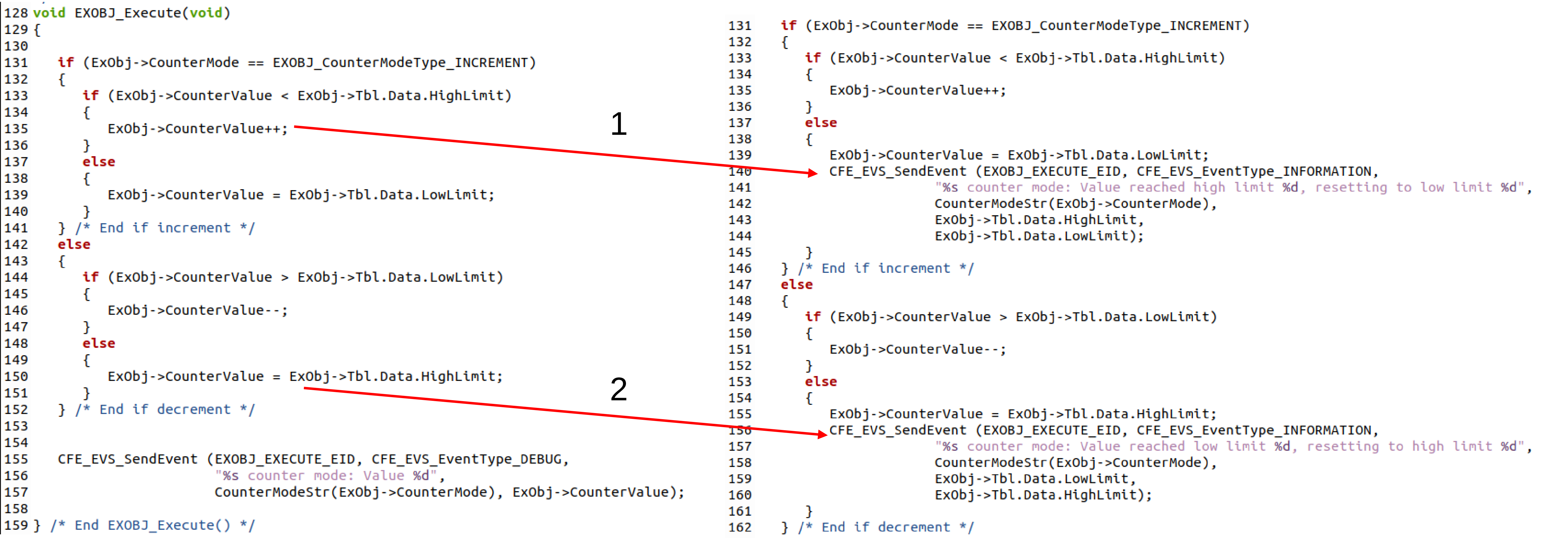


```c
typedef struct
{

   /*
   ** Table parameter data
   */

   EXOBJTBL_Data_t Data;

   /*
   ** Standard CJSON table data
   */

   const char*   AppName;
   bool          Loaded;   /* Has
   uint8         LastLoadStatus;
   uint16        LastLoadCnt;

   size_t        JsonObjCnt;
   char          JsonBuf[EXOBJTBL_
   size_t        JsonFileLen;

} EXOBJTBL_Class_t;
```

Add Event Messages to ExObj Execute()

```
cfsat
├-cfe-eds-framework/
│  ├-cfsat_defs/
│  │  └-cpu1_hello_tbl.json
└-cfs-apps/
   └-hello/
      ├-eds/
      ├-fsw/src/
         └-exobj.c
```

- **Open *./cfs-apps/hello/fsw/src/exobj.c in a text editor***
- **Event message function call**
  - `CFE_EVE_SendEvent(Event ID, Event Type, Format String, Values)`
- **Add events as show below**
  1. Send event when increment mode reaches its high limit
  2. Send event when decrement mode reaches its low limit



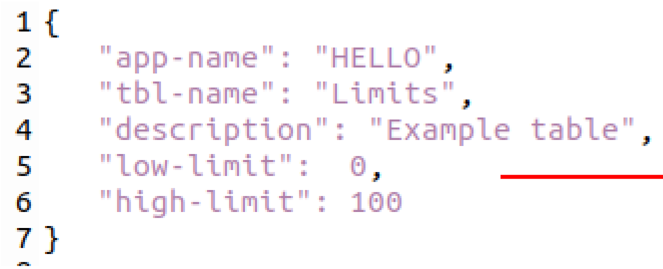
```
void EXOBJ_Execute(void)
{

   if (ExObj->CounterMode == EXOBJ_CounterModeType_INCREMENT)
   {
      if (ExObj->CounterValue < ExObj->Tbl.Data.HighLimit)
      {
         ExObj->CounterValue++;
      }
      else
      {
         ExObj->CounterValue = ExObj->Tbl.Data.LowLimit;
      }
   } /* End if increment */
   else
   {
      if (ExObj->CounterValue > ExObj->Tbl.Data.LowLimit)
      {
         ExObj->CounterValue--;
      }
      else
      {
         ExObj->CounterValue = ExObj->Tbl.Data.HighLimit;
      }
   } /* End if decrement */


   CFE_EVS_SendEvent (EXOBJ_EXECUTE_EID, CFE_EVS_EventType_DEBUG,
                      "%s counter mode: Value %d",
                      CounterModeStr(ExObj->CounterMode), ExObj->CounterValue);

} /* End EXOBJ_Execute() */
```

```
   if (ExObj->CounterMode == EXOBJ_CounterModeType_INCREMENT)
   {
      if (ExObj->CounterValue < ExObj->Tbl.Data.HighLimit)
      {
         ExObj->CounterValue++;
      }
      else
      {
         ExObj->CounterValue = ExObj->Tbl.Data.LowLimit;
         CFE_EVS_SendEvent (EXOBJ_EXECUTE_EID, CFE_EVS_EventType_INFORMATION,
                      "%s counter mode: Value reached high limit %d, resetting to low limit %d",
                      CounterModeStr(ExObj->CounterMode),
                      ExObj->Tbl.Data.HighLimit,
                      ExObj->Tbl.Data.LowLimit);
      }
   } /* End if increment */
   else
   {
      if (ExObj->CounterValue > ExObj->Tbl.Data.LowLimit)
      {
         ExObj->CounterValue--;
      }
      else
      {
         ExObj->CounterValue = ExObj->Tbl.Data.HighLimit;
         CFE_EVS_SendEvent (EXOBJ_EXECUTE_EID, CFE_EVS_EventType_INFORMATION,
                      "%s counter mode: Value reached low limit %d, resetting to high limit %d",
                      CounterModeStr(ExObj->CounterMode),
                      ExObj->Tbl.Data.LowLimit,
                      ExObj->Tbl.Data.HighLimit);
      }
   } /* End if decrement */
```

Change Default Counter Limits

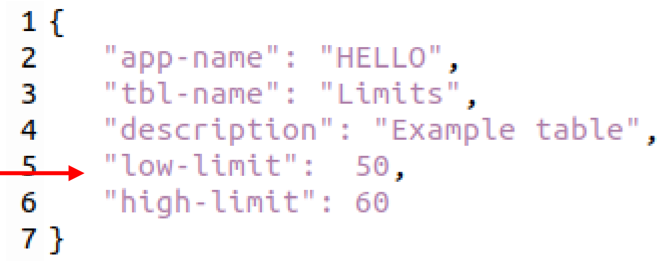
- Open *./cfe-eds-framework/cfsat_defs/cpu1_hello_tbl.json in a text editor*
- **Change low limit from 0 to 50**
- **Change high limit from 100 to 60**

```
cfsat
├-cfe-eds-framework/
│  ├-cfsat_defs/
│  │  └-cpu1_hello_tbl.json
```



```
{
   "app-name": "HELLO",
   "tbl-name": "Limits",
   "description": "Example table",
   "low-limit":  0,
   "high-limit": 100
}
```

1

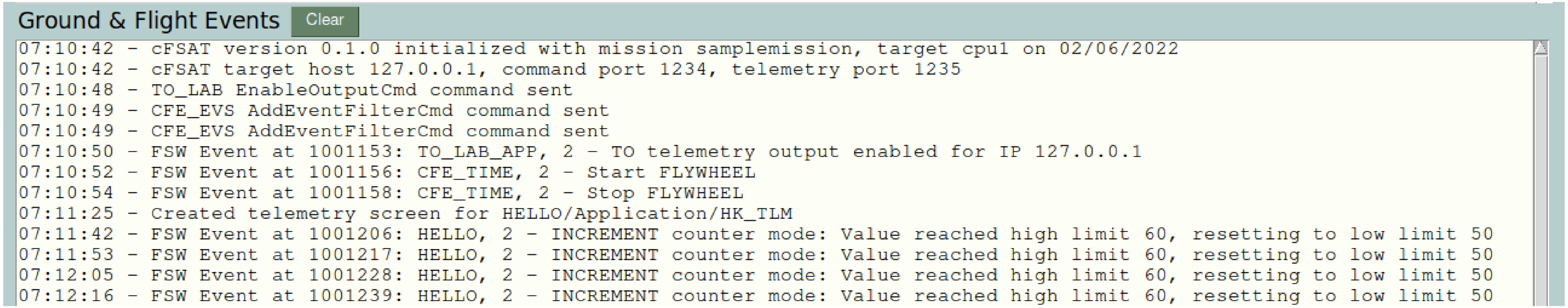


```
{
   "app-name": "HELLO",
   "tbl-name": "Limits",
   "description": "Example table",
   "low-limit":  50,
   "high-limit": 60
}
```

Build and cFS with Updated Example Object

```
cfsat
├-cfe-eds-framework/
  └-build/exe/
          └-cpu1/
            └-cf/
```

1. **Change directory to *./cfe-eds-framework/***
   - `Make install`
2. **Change directory to *./cfe-eds-framework/build/exe/cpu1***
   - `./core-cpu1`
3. **From a different terminal change directory to ./cfsat/gnd-sys/app**
   - `. ./setvars.sh`
   - `Python3 cfsat.py`
   - `cFSAT: cFS Config -> Enable Telemetry`
4. **Try hello's Set Mode Command to verify your changes behave as expected**



```
07:10:42 - cFSAT version 0.1.0 initialized with mission samplemission, target cpu1 on 02/06/2022
07:10:42 - cFSAT target host 127.0.0.1, command port 1234, telemetry port 1235
07:10:48 - TO_LAB EnableOutputCmd command sent
07:10:49 - CFE_EVS AddEventFilterCmd command sent
07:10:49 - CFE_EVS AddEventFilterCmd command sent
07:10:50 - FSW Event at 1001153: TO_LAB_APP, 2 - TO telemetry output enabled for IP 127.0.0.1
07:10:52 - FSW Event at 1001156: CFE_TIME, 2 - Start FLYWHEEL
07:10:54 - FSW Event at 1001158: CFE_TIME, 2 - Stop FLYWHEEL
07:11:25 - Created telemetry screen for HELLO/Application/HK_TLM
07:11:42 - FSW Event at 1001206: HELLO, 2 - INCREMENT counter mode: Value reached high limit 60, resetting to low limit 50
07:11:53 - FSW Event at 1001217: HELLO, 2 - INCREMENT counter mode: Value reached high limit 60, resetting to low limit 50
07:12:05 - FSW Event at 1001228: HELLO, 2 - INCREMENT counter mode: Value reached high limit 60, resetting to low limit 50
07:12:16 - FSW Event at 1001239: HELLO, 2 - INCREMENT counter mode: Value reached high limit 60, resetting to low limit 50
```