# User's Manual

## Start Offset Assistant

abaco
SYSTEMS

## Copyrights

Document Revision:     1.10
Document Date:         8 January 2020

Abaco Systems, Inc.
26 Castilian Drive, Suite B
Goleta, CA 93117
Main          +1 805-965-8000 or +1 877-429-1553
Support       +1 805-883-6097

avionics.support@abaco.com (email)
https://www.abaco.com/products/avionics

## Additional Resources

For more information, please visit the Abaco Systems website at:

www.abaco.com

# Contents

# Introduction

## 1.1 What is Start Offset Assistant?

Start Offset Assistant is a combination of two 32-bit Windows applications named gen_offsets.exe and test_offsets.exe.   The application gen_offsets.exe will process a text file containing multiple transmit channel scheduled message scenarios and generate start offset values for each message entry for use with CEI-x30 and RAR-USB product application programmer's interface functions *ar_define_msg_block* or *ar_define_msg*.   The application test_offsets.exe will process the output text file from gen_offsets.exe and verify the resulting message scenario with your respective ARINC device.

DISCLAIMER: While the gen_offsets.exe application will generate acceptable start offset values for most scheduled message scenarios, there are cases where the complexity of multiple message rate groups on a single channel are beyond the capability of this application to accurately generate valid offset values.

## 1.2 System Requirements

The applications gen_offsets.exe and test_offsets.exe are 32-bit applications that will support the following 32-bit and 64-bit Windows operating systems:

    10/8.1/8/7, Server 2012R1+R2, Server 2008R2, XP (32-bit only)

## 1.3 Finding the Programs

After installing the software distribution, use the Windows Explorer to navigate to the folder with respect to the installed device (CEI-x30 or RAR-USB, respectively):

    \Users\Public\Documents\Condor Engineering\CEI-x30-SW\Utilities\Start Offset Assistant
or
    \Users\Public\Documents\Condor Engineering\AR-STREAM-SW\Utilities\Start Offset Assistant

Here you will find both executables, along with sample input and output text files and report files.

# Program Usage and File Formats

## 2.0 Overview

This chapter describes the file formats for both **gen_offsets.exe** and **test_offsets.exe**.   Both applications will accept a specific input file if the entire filename is specified as a command line parameter in the application invocation.   Sample batch files are included to demonstrate the use of the command line filename option with gen_offsets.exe.

## 2.1 Initial Scheduled Message Scenario Input File

To acquire the start offset values for a transmit channel scheduled message scenario, create an input text file with message definitions based on the C structure type AR_SCHEDULED_MSG_ENTRY_TYPE (see CDEV_API.H for the CEI-x30-SW distribution or SAR_API.H for the AR-STREAM-SW distribution).   Two sample input text files are provided, msg_100K_no_offsets.txt.   Viewing either of these files you will see the required input file format for the **gen_offsets.exe** application, demonstrated as follows:

```
channel = 0, busSpeed = HIGH
channel = 1, busSpeed = LOW
AR_SCHEDULED_MSG_ENTRY_TYPE tx5_command_block[27] = {
/* LABEL                    Message Device   TX    Msg    Start   Transmit Count   Data   */
/*                          Index   Number  Chan   Rate   Offset                          */
/* FaultMaint_EEC_L1,    */ {0,      0,     0,     800,    0,    LOCAL_TX_COUNT, LABEL_001},
/* FaultMaint_EEC_L2,    */ {0,      0,     1,     800,    0,    LOCAL_TX_COUNT, LABEL_002},
                                    . . .
};
```

The first line contains the tokens *channel* and *busSpeed*, together indicating the bus speed of each of the respective transmit channels referenced in the subsequent message entries.   These channel/bus speed entries are not required to be listed in any order but should maintain the spacing format as shown.
The bus speed is used by gen_offsets.exe to determine the duration to transmit individual messages on the respective channel, affecting the applied start offset values. Valid entries for the *busSpeed* token are:

```
    busSpeed = LOW
    busSpeed = HIGH
```
Where LOW indicates a 12.5Kbps bus speed, and HIGH indicates a 100Kbps bus speed.

Channel entries are zero-based, and should be limited to only those transmit channels referenced in the subsequent message entries.

The next line containing the token *AR_SCHEDULED_MSG_ENTRY_TYPE* indicates the beginning of the data structure implementation in which the scheduled message scenario is defined.

Any line following the line with the token *AR_SCHEDULED_MSG_ENTRY_TYPE* that contains the character "{" will be treated as a scheduled message element of the type AR_SCHEDULED_MSG_ENTRY_TYPE, and processed as such.    Comments entered on a line prior to the "{" character are allowed, and do not have to conform to C comment styles, but are not transferred to the output file.    Once the "{" character is encountered, the format should conform to the data structure type AR_SCHEDULED_MSG_ENTRY_TYPE; however, the Transmit Count and Message Data entries are treated as string elements and have no bearing on the calculations for start offset values.    Only the message rate structure member is used from the input file, while the remaining structure members and applied start offset values are passed to the output file for consistency.

## 2.2 Scheduled Message Output File with Offsets

The output file from gen_offsets.exe is called *msg_setup.txt* and utilizes the following format:

```
channel = 0, busSpeed = HIGH
channel = 1, busSpeed = LOW
messageCount=27
AR_SCHEDULED_MSG_ENTRY_TYPE tx5_command_block[27] = {
   {  0,   0,   0,  800,  139,  LOCAL_TX_COUNT,  LABEL_001},
   {  0,   0,   1,  800,  142,  LOCAL_TX_COUNT,  LABEL_002},
                     ...
};
```

While the *channel/busSpeed* and structure array token entries are maintained from the input text file, no comments are passed to the output file by the application gen_offsets.exe.   The *messageCount* entry indicates the total number of message entries processed from the input file, regardless of the value used in the structure array declaration line.   The lines in the output file that encapsulate the structure array can be copied directly into a C source file.

## 2.3 Scheduled Message Output File Verification Log File

The resulting scheduled message scenario as defined in the output file *msg_setup.txt* in its entirety can be verified using your supported ARINC hardware via the application **test_offsets.exe**.    NOTE: the *board* and *transmit channel* structure member values used in the transmit channel scheduled message scenario input

text file must match a supported installed ARINC device on your host system.   Each message will be defined for the respective board's transmit channel using the *rate* and *start offset* element values for each entry, while the transmit count will be programmed as "continuous transmission" (-1) and the message content will be assigned an incrementing value starting at 0.   Reception of the message scenario is implemented via the Internal Wrap feature for the respective receive channel on the board.

When invoked, this application will log any message rate deviations that exceed the 5% tolerance allowed per the ARINC 429 specification to both the console window and to the log file *msg_scheduler_log.txt*. The format of this log file is:

```
Chan   Message   Rate    min dT    max dT    dT>5pct
  0   0x00000000   800    799.992   800.002
  0   0x00000001   800    799.993   800.002
  0   0x00000002   800    799.992   800.002
  0   0x00000003   800    799.992   800.002
  0   0x00000004   800    799.992   800.002
  0   0x00000005   800    799.992   800.002
  0   0x00000006   800    799.992   800.005
  0   0x00000007   800    799.992   800.005
  1   0x00000008   400    399.995   400.006
  1   0x00000009   100     99.990   100.004
  1   0x0000000A   100     99.991   100.004
  1   0x0000000B   400    399.994   400.008
  1   0x0000000C   400    399.994   400.007
```

The message value indicates the entry in the input file (*msg_setup.txt*) and has no bearing on any message value supplied for the *data* structure member for the respective array element.    If rate skew in excess of 5% of the message rate is encountered by the respective receive channel during transmission of the scheduled message scenario, each instance of an out-of-bounds duration between subsequent messages (up to 10) will be noted beneath the "dT>5pct" column.