

This Application Note applies to the Unidrive SP, Commander GP20 and Commander SK/LogicStick

Designing and Coding an Application Specific Control Word

Using SyPT Lite

Scope – Many application requirements call for either serial or process bus control of one or more Drives from some Master Process Controller. Master Process Controllers are typically implemented via a Programmable Logic Controller (PLC), a Process Control Computer (PCC), or a Unidrive SP second processor (SM-Apps or a SM-Apps Lite).

It is both common and efficient to use a “collection” of command bits in the format of a 16-bit word as an exchange medium for the “command events” between the Master Process Controller and each of the individual Drives. Such a “collection” is commonly referred to as a Control Word.

The Unidrive SP and the Commander SK feature (nearly) identical support for a “native” control word, as well as the ability to “arbitrate” between using the drive terminal control connections (terminal control) and using the control word for serial / process bus control.

The “native” control word of the SP or the SK is suitable if only simple “auto/manual”, “run”, “inch/jog”, and “direction” selection is required. More advanced applications can (often) be optimized by encoding additional command events on an “application specific” control word, and coding additional logic to “handle” these extensions in the Drive.

This Application note describes this technique using the features available in the Control Techniques SyPT Lite development environment. These same approaches can also be used in the (more powerful) SyPT Pro development environment.

Assumptions

This application note assumes that the reader is already proficient with the Unidrive SP or the Commander SK/LogicStick, including the Advanced Parameters of the product.

This application note assumes that the reader already has [SyPT Lite](#) installed and activated on a suitable PC running Windows 98SE or better. This includes having a serial port and CT Communication cable to connect to the product. CT Soft is also very useful, especially with the Commander SK/ LogicStick combination.

This application note assumes the reader is proficient with creating, editing and saving a SyPT Lite program.

This application note assumes the reader is proficient with opening, modifying and saving an existing SyPT Lite program.

This Application note will illustrate the logic initially on a Unidrive SP and running on a SM-Application Lite module. Internally, most CT program development is initially done on the SP fitted with either a SM-Application module, or a SM-Application Lite module. This template is then moved to the target platform via change of the project properties, and a re-compile. Note that the code provided here will run (after re-target and re-compilation) on the SP/SM-Apps, SP/SM-Apps Lite, SP onboard, and SK/LogicStick.

Table of Contents - Overview

This Application note is the first in a related series, and will describe the design and construction of a alternative control word that encodes the “*minimum*” command events native to these drives, in the most efficient format possible. The intent here is to make the most bit positions available for encoding additional command events. This logic is then used as the starting point for the next application note.

The second Application Note of this series will describe a simple extension of this “minimal” alternative control word, creating a true “*application specific*” control word. This application note will also include some suggestions on minimal documentation requirements.

The third Application Note of this series of this application note will discuss the watchdog feature of the “*native*” control word (bit 14) and how one can include a “**serial link loss**” feature with an “application specific” control word.

Step 1

Design and code a “reusable” starting point

The strategy is to select an alternative destination for an Application Specific control word, and code logic in the drive’s programmable features to implement this new behavior. The author has performed this step, and the solution is presented in this application note.

Register usage

Menu 6

Bit Parameter #6.15 – Drive Enable Status / Control (usually used as a software enable / disable)

Bit Parameter #6.29 – Drive Hardware Enable Status

Bit Parameter #6.43 – Control Word Enable

Register Parameter #6.42 – Drive (native) Control Word

Menu 18 is chosen for use because it includes 16-bit general-purpose registers on both the Unidrive SP and the Commander SK that can be used to read and write integer (register) values.

The first read/write parameter in menu 18, Parameter #18.11, is chosen as the location for the new (alternative) control word.

The last read-only parameter in menu 18, Parameter #18.10 is chosen as the location for a needed working (temporary) register.

Register Parameter #18.11 – Alternative Control Word Destination

Register Parameter #18.10 – Temporary working register for the Control Word Image

Implemented in **CWRemapT.dpl**. **Click here to download ----> [RemapExample](#)**

Target Platform - UnidriveSP fitted with an SM-Application module

The Illustration on the next page documents the major forms encountered in the template version of the program for the UnidriveSP.

(* Control Word Enable hard-wired here *)

(* Get and solve input "Use Control Word" *)

(* get and solve input "Rev/Fwd Select" *)

(* Get and solve input "US Inch", interlocked with "not-Run" *)

(* Get and solve "Run", Interlocked with "not-US Inch" *)

(* Get and solve "Run", Interlocked with "not-US Inch" *)

(* Conditionally update the Control Word with decoded content *)

(* Set control word to nominal do-nothing" value. SK needs the write to #6.15, SP does not *)

Properties

Title:

Version:

Author:

Company:

Target:

Task:

Note: All rungs will execute in the selected Task.

☒ Embed source file

OK Cancel Help

Description

Aliases

Aliases allow you to define your own names for drive parameters.

For example your application may use parameter #18.20 to hold the state of a switch. To make your diagram more readable you can define the alias SWITCH for #18.20. SyPT Lite will then use SWITCH wherever #18.20 appears in the diagram.

Existing Aliases:

Alias	Parameter
CW_TMP_REV_F...	#18.10.04
CW_TMP_RUN	#18.10.05
CONTROL_WORD	#06.42
ALT_TEMP_CTRL...	#18.10
CW_TMP_AUTO...	#18.10.07
CTRL_WORD_ENBL	#06.43
USE_ALT_CW	#18.11.00
HW_DRIVE_ENAB...	#06.29
CW_TMP_ENBL	#18.10.00
ALT_CW_RUN	#18.11.01
ALT_CW_REV_FW...	#18.11.02
CW_TMP_JOG	#18.10.02
ALT_CW_US_INCH	#18.11.03

New... Delete... Change...

Node description

This is an example program can implement both a remap and a re-define of the Control Word of the Drive. This program is designed to compile for both SP (onboard, SM-App and SM-App Lite) and SK (with logic stick).

This logic can be used to, relocate, and/or define combinations of drive's command bits, along with other drive control bits. Provides most of the utility of a custom control word, and retains the original strength of the product's standard control word.

A "servo enable" re-define would assert both the "soft enable" and "run", at "drive enable" input active.

AltControlWord% 00 Use control word
AltControlWord% 01 Run
AltControlWord% 02 Rev/Fwd
AltControlWord% 03 US Inch

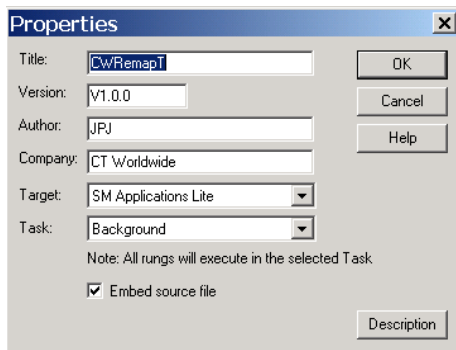
Resource usage:
#18.10 is a temp.
#18.11 is the Alternate Command Word. This is 16 bits on both SP and SK.
#6.42 is the Command Word
#6.43 is the Command Word Select

OK Cancel Help

SyPT Lite Program CwremapT.dpl

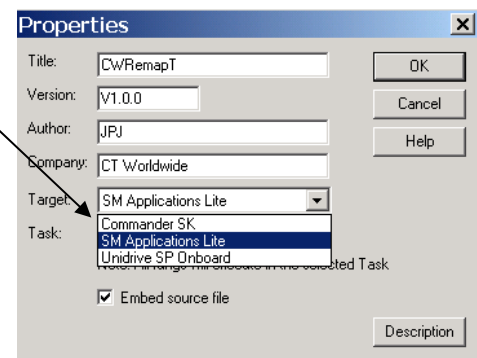
Deriving a SK version from CWRemapT.dpl

1. Open the template program CWRemapT.dpl Obtain from web – click---> [RemapTemplate](#)
2. From the main menu, select File....Diagram Properties....



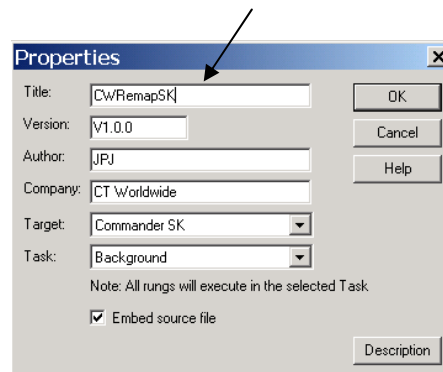
The Properties dialog box for CWRemapT.dpl. The Title field contains 'CWRemapT'. The Version field contains 'V1.0.0'. The Author field contains 'JPJ'. The Company field contains 'CT Worldwide'. The Target dropdown menu is set to 'SM Applications Lite'. The Task dropdown menu is set to 'Background'. A note states: 'Note: All rungs will execute in the selected Task'. The 'Embed source file' checkbox is checked. There are buttons for OK, Cancel, Help, and a Description button.

3. From the Properties Target: dropdown select Commander SK



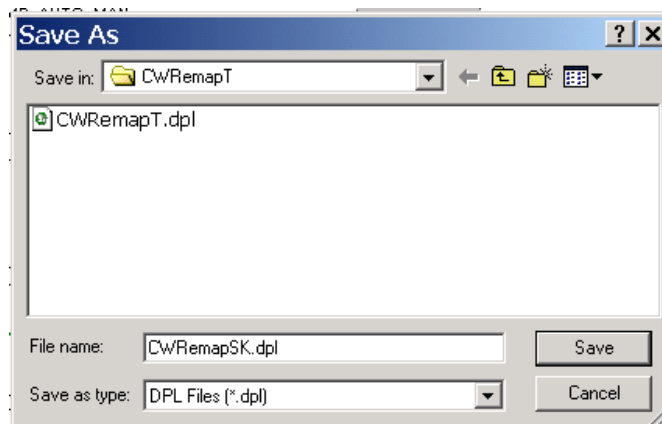
The Properties dialog box for CWRemapT.dpl with the Target dropdown menu open. The dropdown menu shows three options: 'SM Applications Lite', 'Commander SK', and 'Unidrive SP Onboard'. An arrow points from the text 'From the Properties Target: dropdown select Commander SK' to the 'Commander SK' option in the dropdown menu.

4. Modify the Properties Title: to the final title for the program, in this case “CWRemapSK”



The Properties dialog box for CWRemapSK.dpl. The Title field contains 'CWRemapSK'. The Version field contains 'V1.0.0'. The Author field contains 'JPJ'. The Company field contains 'CT Worldwide'. The Target dropdown menu is set to 'Commander SK'. The Task dropdown menu is set to 'Background'. A note states: 'Note: All rungs will execute in the selected Task'. The 'Embed source file' checkbox is checked. There are buttons for OK, Cancel, Help, and a Description button. An arrow points from the text 'Modify the Properties Title: to the final title for the program, in this case “CWRemapSK”' to the Title field.

5. From the main menu, select File....Save As....
6. Edit the File Name to the final File name for the program, in this case CWRemapSK.dpl
7. From the Save As form, left-click on the Save button to save your work



8. From the Main menu, left-click on the Drive... Compile Diagram to prepare the logic for loading.
9. If not on line with the SK already, go on line now
10. From the Main menu, left-click on the Drive...Download Program to Drive
11. Done

Author: **Jim Jeffers**
 (716)-774-1193

e-mail : <mailto:jim.jeffers@emersonct.com>