Alarm Clock V5
Module Application Guide

Description

This module enables alarm clock functionality on any 2-series processor. Most programmers use it to trigger an audio source to wake the user up to music. The module is full featured with functions such as snooze, 12 and 24 hour time formats as well as enables for days of the week that the alarm is active. The module also buffers your room volume so you can ramp up the volume at a selectable rate when the alarm goes off. All values can be stored in RAM so reloading your program will not wipe out your users settings. The module also has outputs that can be tied to other systems such as lights or shades for non-audio applications.

Be sure to check out our family of RFC Clocksync modules that synchronize your processors clock with internet time servers! http://www.thecontrolworks.com/clock.htm

Compatibility			Processor Requirements	
☺ 2-Series Compatible	🚫 NOT CNMSX Compatible	🚫 NOT System Builder Compatible	🚫 Ethernet NOT NEEDED	🚫 Compact Flash NOT NEEDED

Module Application

Multiple instances of this module may be used in the same program.

Should you be using this module for something other than triggering audio you can ignore the alarm source to and from signals on the module.

The module can be simply inserted in your program between your room logic and your PAD8A or BIPAD logic. When the alarm is not in use the module passes all the audio distribution signals through the module.

Be sure to connect the (to_ram) signals to the ARAM symbol in your program. We exposed these signals to prevent having to put an ARAM signal in the module, thus running the risk of changing the ARAM values in your program if you moved the Alarm clock module.

Signal And Parameter Descriptions

Bracketed signals such as “[signal_name]” are optional signals

DIGITAL INPUTS

[alarm_enable] .... pulse to enable alarm functionality

[alarm_disable] .... pulse to disable alarm functionality

alarm_off .... pulse to turn alarm off

alarm_disable_leave_source_playing .... pulse to setup the module so the audio source will remain on after the alarm is triggered and the user then hits “alarm_off”

alarm_disable_turn_room_off .... pulse to setup the module so when the “alarm_off” is pulsed that the room audio is forced off

alarm_snooze .... pulse to snooze and then retrigger source select after “snooze_time” (see parameters) expires

alarm_hr+/- .... pulse to increment/decrement the alarm hour set time

alarm_min+/- .... pulse to increment/decrement the alarm minute set time

alarm_pm .... pulse to define the current alarm set time as PM

alarm_am .... pulse to define the current alarm set time as AM

alarm_12/24 .... pulse to toggle the alarm display between the 12 or 24-hour format

[alarm_no_source] .... pulse to clear source selection for triggering

[alarm_source1…8] .... pulse to select which source to trigger on the alarm event

[source1-8_from_program] .... connect to logic in your program that typically would connect to your PAD8A/BIPAD module or room logic

[room_off_from_program] .... connect to your room off press logic in your program

monday_only…all_days .... enable **only one** of these inputs with a “1” to tell the module which days of the week the alarm should be enabled. Place a “0” (zero) on the other nine unused inputs. Alternatively you can drive all 10 inputs with an interlock to allow the user to select which day(s) are enabled.

ANALOG INPUTS

[volume_from_module] .... tie to the analogs from your PAD8/BIPAD8 module volume output or room logic

alarm_day(to_ram) – alarm_disable_mode(to_ram) ...... tie these analog to an ARAM symbol in your program so the clock settings persist through a program reset or reboot

SERIAL INPUTS

tod$ .... tie to clock driver output

DIGITAL OUTPUTS

alarm_clock_on_fb .... high when alarm clock is enabled

alarm_clock_off_fb .... high when alarm clock is disabled

[alarm_disable_leave_source_playing_fb] .... high to indicate when the module will leave the audio source on once the alarm is turned off

[alarm_disable_turn_room_off_fb] .... high to indicate when the module will turn off the room audio when the alarm is turned off

alarm_trigger_pulsed .... pulses high whenever alarm triggers (goes off)

alarm_trigger_fb .... goes high when alarm triggers, goes low when alarm is disabled, turned off or snoozed

alarm_snooze_fb .... high when the snooze has been activated

alarm_pm_fb .... high when the alarm time entered is for PM

alarm_am_fb .... high when the alarm time entered is for AM

[alarm_source1..8_fb] .... high to indicate which source has been selected for the trigger

[alarm_trigger_source_1..8] .... pulses whenever alarm triggers and appropriate source is selected. Tie to your PAD8A/BIPAD module source input or room logic.

[room_off_to_module] .... pulses when room off is selected in your program logic. Tie to room off on your PAD8A/BIPAD module or room logic.

[alarm_volume_busy] .... high when the volume level is ramping after the alarm goes off

ANALOG OUTPUTS

[volume_to_pad8-bipad8] .... tie to volume analog input on the PAD8A or BIPAD8

SERIAL OUTPUTS

alarm_display$ .... clock display for setting the alarm time. This could be 12:00:00 or just 12:00 depending on the "display_seconds" parameter. (see below)

PARAMETERS

snooze_time .... time (in seconds) to wait once "alarm_snooze" is pulsed before retriggering the alarm again

module_ramp_time .... time (in seconds) that your volume ramps in your room logic for the PAD8A/BIPAD8

volume_ramp_time .... time (in seconds) that you want the volume to ramp up on the selected source once the alarm goes off.

Note: The above two ramp times added together will equal the total time between the alarm trigger and the volume hitting the end of the ramp.

display_seconds .... allows you to select whether the seconds are displayed in the "alarm_display" string.

Support

This module is supported by ControlWorks Consulting, LLC. Should you need support for this module please email support@controlworks.com or call us at 440-449-1100. ControlWorks normal office hours are 9 AM to 5 PM Eastern, Monday through Friday, excluding holidays.

Before calling for support, please ensure that you have loaded and tested operation using the included demonstration program and touchpanel(s) to ensure that you understand the correct operation of the module. It may be difficult for ControlWorks to provide support until the demonstration program is loaded.

Updates, when available, are automatically distributed via Email notification to the address entered when the module was purchased. In addition, updates may be obtained using your username and password at http://www.thecontrolworks.com/customerlogin.aspx

Distribution Package Contents

The distribution package for this module should include:

Alarm_Clock_v5.umc	Crestron User Module
Alarm_clock_trigger_v5.usp	SIMPL+ file used within the module
Alarm_clock_trigger_v5.ush	SIMPL+ header file
Alarm_Clock_Demo_Xpanel_v5.vtp	Demo touchpanel for Xpanel
Alarm_Clock_Demo_Program_v5.smw	Demo program for PRO2 processor
Alarm_Clock_v5_Help.pdf	This help file

Revision History

V5 jim@controlworks.com, gary@controlworks.com, lincoln@controlworks.com 2009.4.2

Added: Inputs to allow the programmer to select whether or not the module shuts off the audio once the alarm is turned off

Fixed: An issue where if the snooze was triggered during a volume ramp erratic volume behavior would occur once the snooze expired and the alarm was re-triggered

V4 (internal) jim@controlworks.com

Added: The ability to show or hide the seconds in the time set string output

Fixed: The alarm_triggered_pulsed not triggering after a snooze

V3 (internal) caleb@controlworks.com

Changed: moved analogs to the defargs so they could be connected to ARAM outside of the module.

V2 jim@controlworks.com 2006.10.31

Added: Changed day enables from a parameter to digital inputs so enable could be programmatically driven.

Added: New logic so module could be dropped into the audio logic path without the need for OR's or BUFFER's.

Added: Ability to pass room volume through module for timed ramp up

Added: New help file format

Added: New TPS-4500 UI

Changed: Made all values non-volatile.

Fixed: Potential issue where a buffer triggering the audio logic would act as a transition gate

V1 bob@controlworks.com

Initial release

Development Environment

This version of the module was developed and tested on the following hardware and software. Different versions of hardware or software may or may not operate properly. If you have questions, please contact us.

Hardware	**Firmware Version**
Crestron PRO2 Processor	3.155.1240
Crestron BIPAD	1.08
Software	**Software Version**
Crestron SIMPL Windows	2.10.32
Crestron Vision Tools Pro-e	3.9.23
Crestron Database	20.04.006.00
Device Database	20.06.016.00
Crestron Symbol Library	589

ControlWorks Consulting, LLC Module License Agreement

Definitions:
ControlWorks, *We*, and *Us* refer to ControlWorks Consulting, LLC, with headquarters located at 701 Beta Drive, Suite 22 Mayfield Village, Ohio 44143-2330. *You* and *Dealer* refer to the entity purchasing the module. *Client* and *End User* refer to the person or entity for whom the Crestron hardware is being installed and/or will utilize the installed system. *System* refers to all components described herein as well as other components, services, or utilities required to achieve the functionality described herein. *Module* refers to files required to implement the functionality provided by the module and may include source files with extensions such as UMC, USP, SMW and VTP. *Demo Program* refers to a group of files used to demonstrate the capabilities of the Module, for example a SIMPL Windows program and VisionTools Touchpanel file(s) illustrating the use of the Module but not including the Module. *Software* refers to the Module and the Demo Program.

Disclaimer of Warranties
ControlWorks Consulting, LLC software is licensed to You as is. You, the consumer, bear the entire risk relating to the quality and performance of the Software. In no event will ControlWorks Consulting, LLC be liable for direct, indirect, incidental or consequential damages resulting from any defect in the Software, even if ControlWorks Consulting, LLC had reason to know of the possibility of such damage. If the Software proves to have defects, You and not Us must assume the cost of any necessary service or repair resulting from such defects.

Provision of Support
We provide limited levels of technical support only for the most recent version of the Module as determined by Us. We do not provide support for previous version of the module, modifications to the module not made by Us, to persons who have not purchased the module from Us. In addition, we may decline to provide support if the Demo Program has not been utilized. We may withdraw a module from sale and discontinue providing support at any time and for any reason, including, for example, if the equipment for which the Module is written is discontinued or substantially modified. The remainder of your rights and obligations pursuant to this license will not be affected should ControlWorks discontinue support for a module.

Modification of Software
You may not decrypt (if encrypted), reverse engineer, modify, translate, disassemble, or de-compile the Module in whole or part. You may modify the Demo Program. In no event will ControlWorks Consulting, LLC be liable for direct, indirect, incidental or consequential damages resulting from You modifying the Software in any manner.

Indemnification/Hold Harmless
ControlWorks, in its sole and absolute discretion may refuse to provide support for the application of the Module in such a manner that We feel has the potential for property damage, or physical injury to any person. Dealer shall indemnify and hold harmless ControlWorks Consulting LLC, its employees, agents, and owners from any and all liability, including direct, indirect, and consequential damages, including but not limited to personal injury, property damage, or lost profits which may result from the operation of a program containing a ControlWorks Consulting, LLC Module or any component thereof.

License Grant
Software authored by ControlWorks remains the property of ControlWorks. ControlWorks grants You the non-exclusive, non-transferable, perpetual license to use the Software authored by ControlWorks as a component of Systems programmed by You. This Software is the intellectual property of ControlWorks Consulting, LLC and is protected by law, including United States and International copyright laws. This Software and the accompanying license may not be transferred, resold, or assigned to other persons, organizations or other Crestron Dealers via any means.

The use of this software indicates acceptance of the terms of this agreement.