Application note

L99LDLH32 – PWM synchronization among several devices

Introduction

When operating multiple devices on the same PCB, it is useful to synchronize these devices so that they have the same PWM frequency. In addition, the start of the PWM cycle of each device can be shifted from each other.

This feature, together with the gradual delay function (delay between individual channels of the device), allows the inrush current from the preregulator to be "smoothed" when multiple channels are activated at the same time or when operating continuously in PWM mode. This helps to minimize the voltage drops/peaks on the preregulator output, caused by its limited transient response capability.

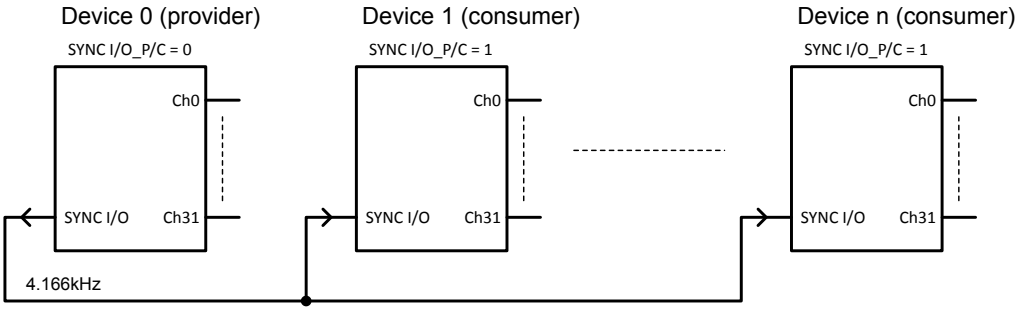
This is particularly important when you use the SEPIC topology, where the bandwidth of the control loop is limited by the presence of a right half zero in its transfer function. Smoothing the inrush current then allows the use of a smaller output capacitor without the risk of excessive voltage drops/false open load detections.

1 PWM synchronization between devices

1.1 Device connection and configuration

The PWM frequency and phase relation of several devices can be synchronized versus each other through the SYNC_I/O pin. There is one dedicated bit (SYNC I/O_P/C – see FTP/RAM memory map in Reference documents) to configure the device in clock provider [1] or clock consumer [0] mode. In provider mode, the device clock frequency (divided to 4.166 kHz) is sent on the SYNC_I/O pin. In consumer mode, the clock frequency on the SYNC_I/O line is used to synchronize its own clock.

Figure 1. PWM synchronization between devices via SYNC_I/O pin



The phase relation among several devices can be configured by PHASE_DEV bits (3-bits - see FTP/RAM memory map – see Reference documents).

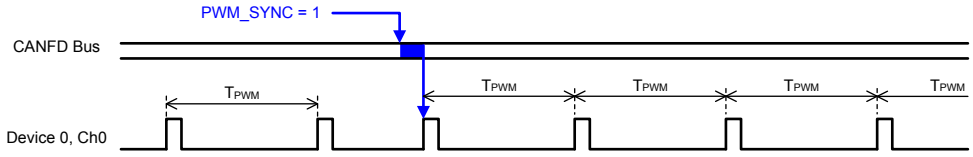
Note: The PHASE_DEV value can be changed only via FTP programming (row 12). In the RAM (Configuration register #1, address 15h) it is read only:

Table 1. PHASE_DEV values

PHASE_DEV [2÷0]			PWM_PHASE_SHIFT [µs]
Bit 2	Bit 1	Bit 0	
0	0	0	0
0	0	1	15
0	1	0	30
0	1	1	45
1	0	0	60
1	0	1	75
1	1	0	90
1	1	1	105

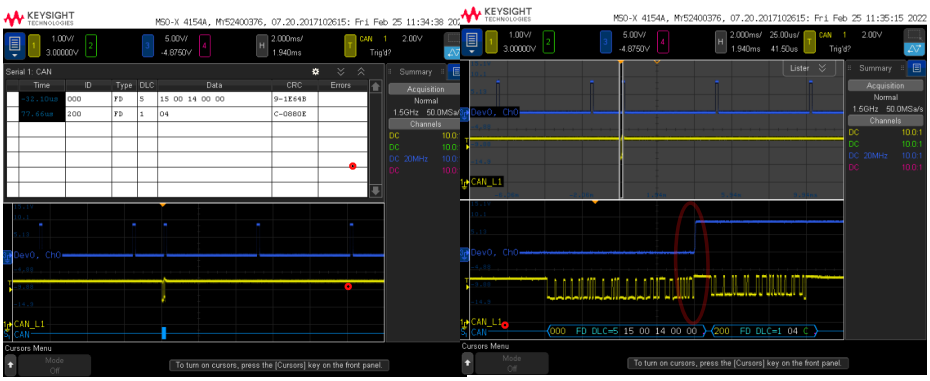
The PHASE_DEV value determines the initial offset of the internal PWM counter. This offset (time delay) is applied after a power on reset or after the user sends the PWM_SYNC=1 reset command. The PWM_SYNC is a bit n.18 in the configuration register #1 (address 15h). When this bit is set to “1”, the PWM counter is reset after the time offset defined by PHASE_DEV and the bit is automatically cleared. The effect of this reset command and the specific PHASE_DEV value can be seen in the following figures. The following figure shows the effect of the reset command with PHASE_DEV value set to 0. In this case the PWM counter is reset immediately after receiving the command, so the new PWM cycle is immediately started.

Figure 2. Effect of the PWM_SYNC=1 command with PHASE_DEV = 0



The previously explained case is demonstrated on a real device in the following figure that shows the CAN bus communication (yellow plot) and the Output 0 configured to 3% duty cycle (blue plot). The PWM counter is reset just after receiving PWM_SYNC=1 reset command (see rising edge of the PWM cycle at the end of the CAN command on detailed screenshot):

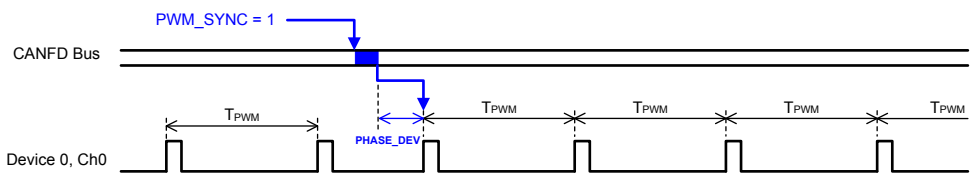
Figure 3. Demonstration of the PWM_SYNC=1 @ PHASE_DEV = “000”



When sending the PWM_SYNC=1 command using register 15h, take care also about other values configured by this register (like SYNC_I/O_M/S, PWM frequency, OUT_DELAY etc.). In case the application SW wants to reset the PWM counter but do not know these other settings (for example when default values, configured through FTP, are used), it is possible to first execute a read command and after sending a write command with the same content with the PWM_SYNC bit changed to "1".

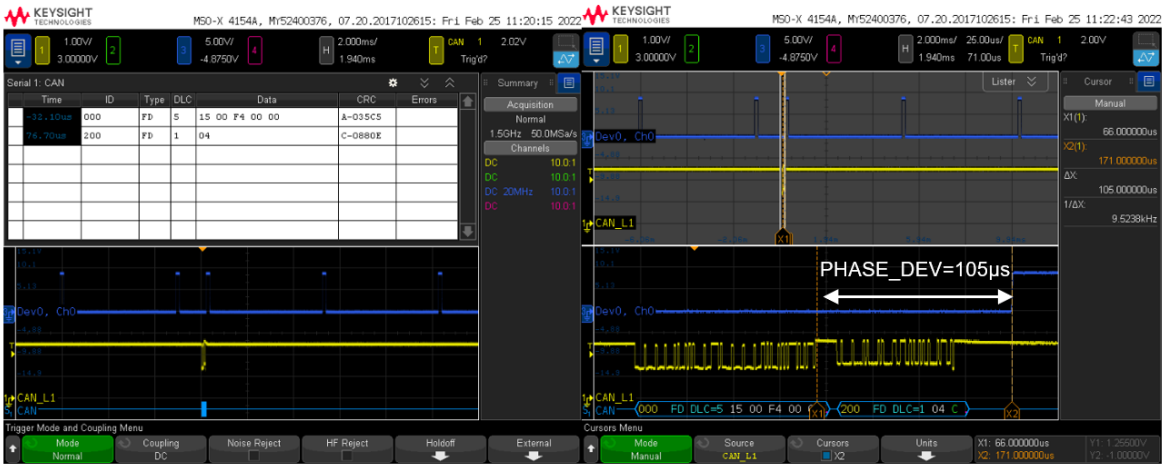
The following figure shows the effect of PWM_SYNC=1 when PHASE_DEV is set to a non-zero value. As shown in the figure, the PWM counter is reset after the configured PHASE_DEV delay time, so the start of the new PWM cycle is delayed from this reset command.

Figure 4. Effect of the PWM_SYNC=1 command with non-zero PHASE_DEV value



The previously explained case is demonstrated on a real device in the following figure. The PHASE_DEV is set to maximum value of 105 µs in this example, so the PWM counter is reset with delay of 105 µs after receiving PWM_SYNC=1 reset command (see detailed screenshot in the following figure):

Figure 5. Demonstration of the PWM_SYNC=1 @ PHASE_DEV = “111” (105 µs)



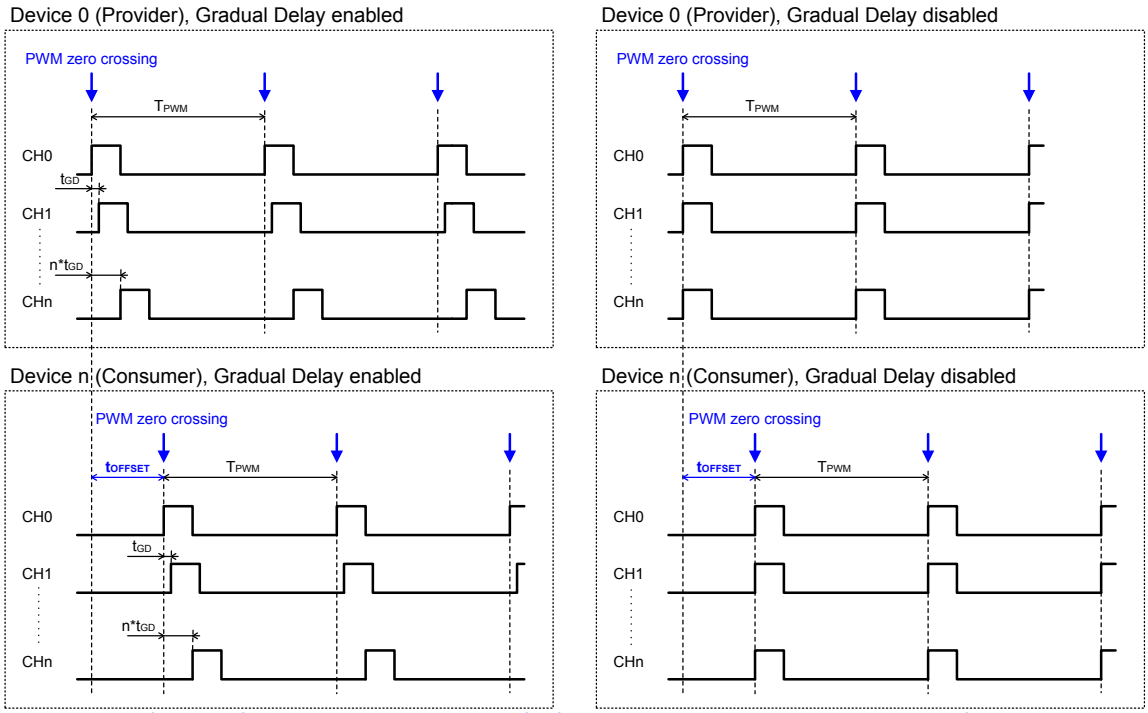
1.2 Synchronization sequence

As explained in the previous chapter, the devices synchronized via the SYNC_I/O pins operate at exactly the same PWM frequency. However, even if the devices are synchronized, the phase relation between them is not accurately defined after system startup (Power- on Reset).

The reason for this initial phase inaccuracy is the fact that the device logic is supplied from its own V3V3 voltage with an external decoupling capacitor, so the time to reach the reset threshold (= starting the PWM counter and applying configured PHASE_DEV) may vary from device to device due to the tolerance of the capacitor value and the tolerance of POR threshold. Also, blocking consumer devices at the provider's frequency is not instantaneous, so all devices run on their own frequency, with some tolerance, for some time.

It causes sometimes delay between devices, shown as "t_{OFFSET}" in the following figure. In case the configured PHASE_DEV value is 0 on all devices, the t_{OFFSET} represents only the above-mentioned tolerances. If a non-zero PHASE_DEV is configured, the t_{OFFSET} is a sum of configured PHASE_DEV and offset due to tolerances.

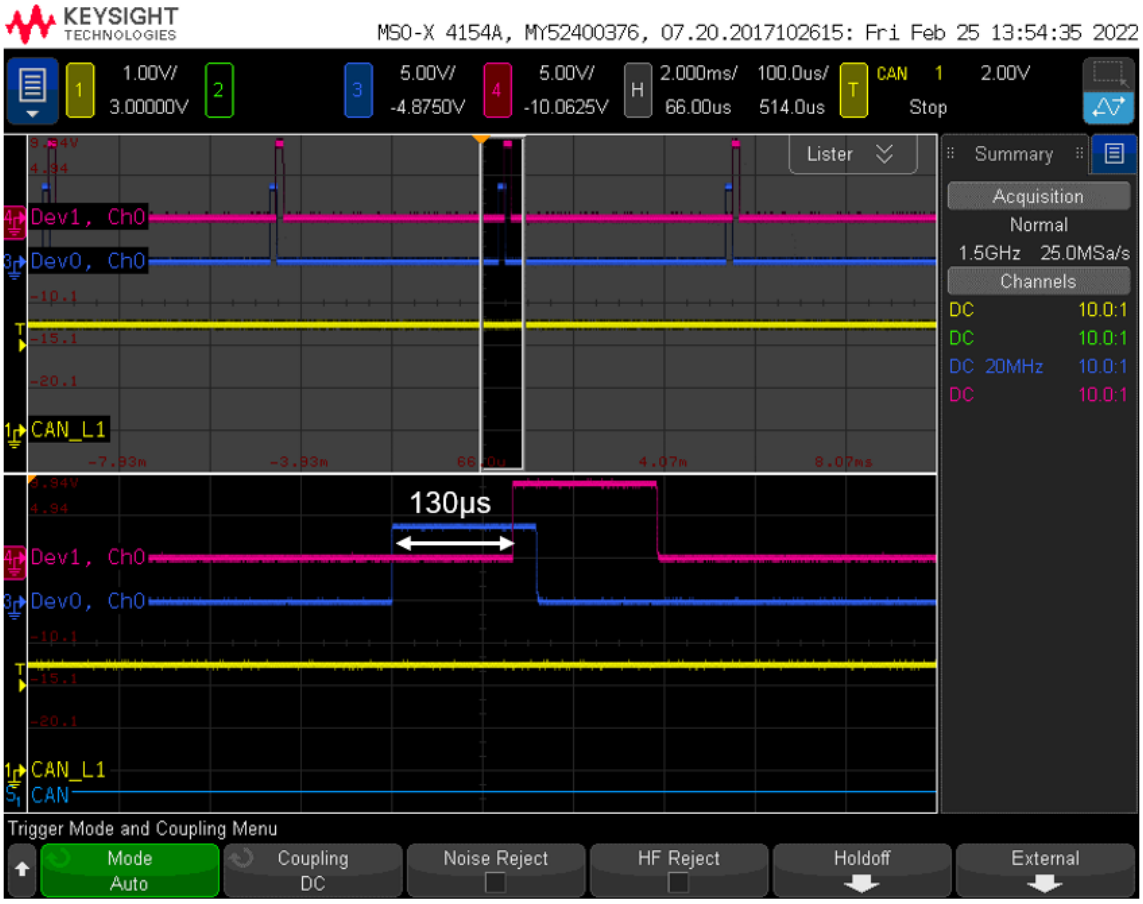
Figure 6. PWM synchronization between devices via SYNC_I/O pin



t_{OFFSET} is defined by PHASE_DEV value and delay between PWM_SYNC=1 commands, resetting the PWM counter in Provider and Consumer device

This initial time offset between devices is also demonstrated in the following figure. In this case, the two L99LDLH32 devices are synchronized via the SYNC_I/O pins and PHASE_DEV is set to 0 on both devices. The blue plot shows PWM signal on CH.0 of device 0 (3.3% duty cycle) and the red plot shows PWM signal on CH.0 of device 1. The measured time offset is 130 µs:

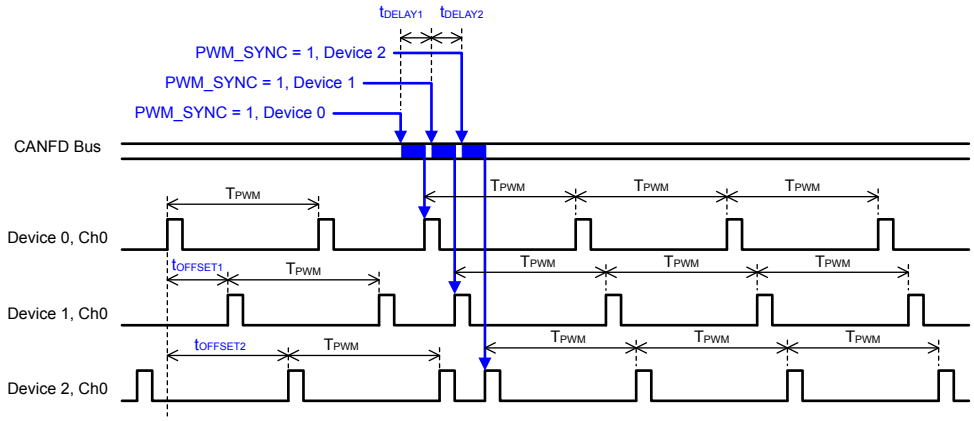
Figure 7. **Initial phase shift between 2 devices after POR**



1.2.1 Synchronization sequence–example 1

In order to eliminate this not very well-defined time offset, a synchronization sequence, writing PWM_SYNC bit of each device to “1”, can be applied. An example of such synchronization sequence is depicted in the following figure. This example shows three synchronized devices whose initial (not well defined) time offset is $t_{OFFSET1}$ and $t_{OFFSET2}$. Then the PWM_SYNC=1 command is sequentially applied to device 0, device 1, and device 2. Since the PHASE_DEV value is set to 0 for all devices, the PWM counter of a particular device is reset just after the command is received. This means that the phase shift between devices after the synchronization sequence is defined by the time delay of the sent commands, t_{DELAY1}, and t_{DELAY2}.

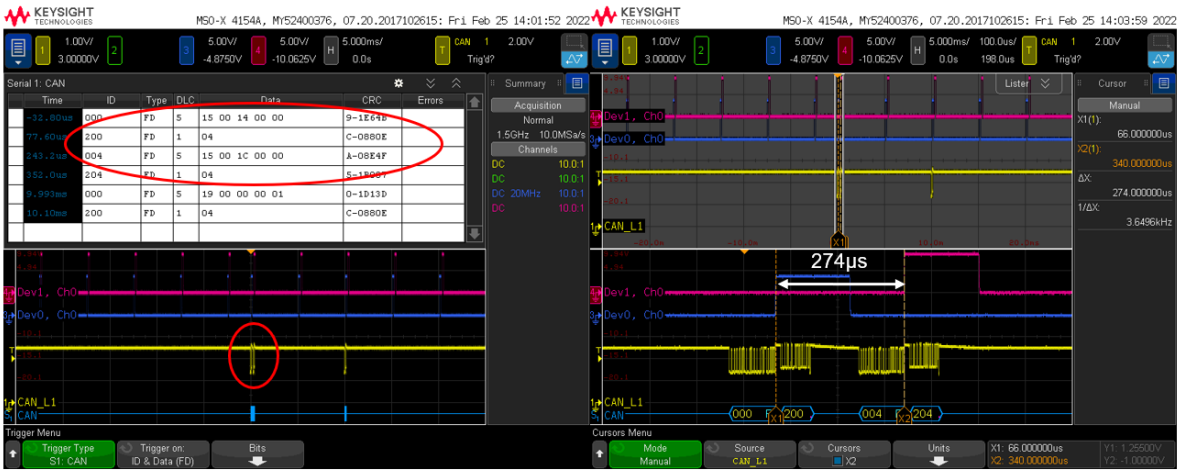
Figure 8. Synchronization sequence example 1)



Note: The outputs on the previous figure are active for illustration purposes only, but it is not necessary. Normally, the synchronization sequence should be sent once during application initialization before the outputs are activated.

The synchronization sequence in example 1 is also demonstrated on real devices in the following figure (in this case there are only two devices). The phase shift between devices is defined by sending 2 PWM_SYNC=1 commands with delay of 274µs:

Figure 9. Demonstration of example 1)

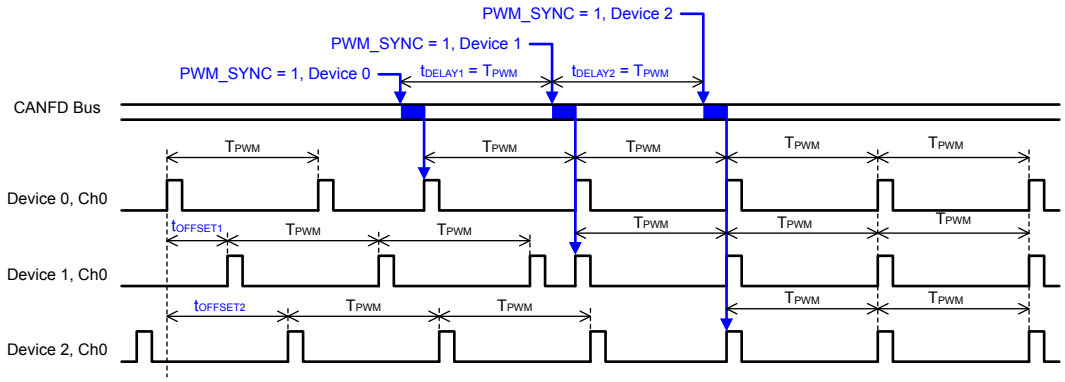


1.2.2 Synchronization sequence – example 2

As explained in the previous example, the PWM_SYNC=1 command can only be sent to one device at a time, so a sequence of several commands is required to reset all devices. Hence, it is not possible to reset all devices at once. But then, how to achieve such synchronization when the PWM zero crossings of all devices are aligned (without any phase shift) or defined only by the PHASE_DEV value programmed in FTP?

The answer to this question is shown in the following figure. The principle is to send a sequence of individual PWM_SYNC=1 commands with a delay of exactly 1 PWM period. In this way, a zero phase shift between devices or a phase shifts defined only by PHASE_DEV can be achieved.

Figure 10. Synchronization sequence example 2)



The accuracy of this adjustment depends on the accuracy of the delay between PWM_SYNC=1 commands (the error is defined by the difference between the applied t_{DELAY} and the actual TPWM). If the error needs to be minimized, the SYNC_I/O line frequency can be measured by the microcontroller to determine the actual PWM period.

The example 2 is also demonstrated on real devices in the following figure with PHASE_DEV=0 on both devices, respectively in the following figure, with PHASE_DEV=0 on Device 0 and PHASE_DEV=60µs on Device 1.

Figure 11. Demonstration of example 2) – PHASE_DEV=0

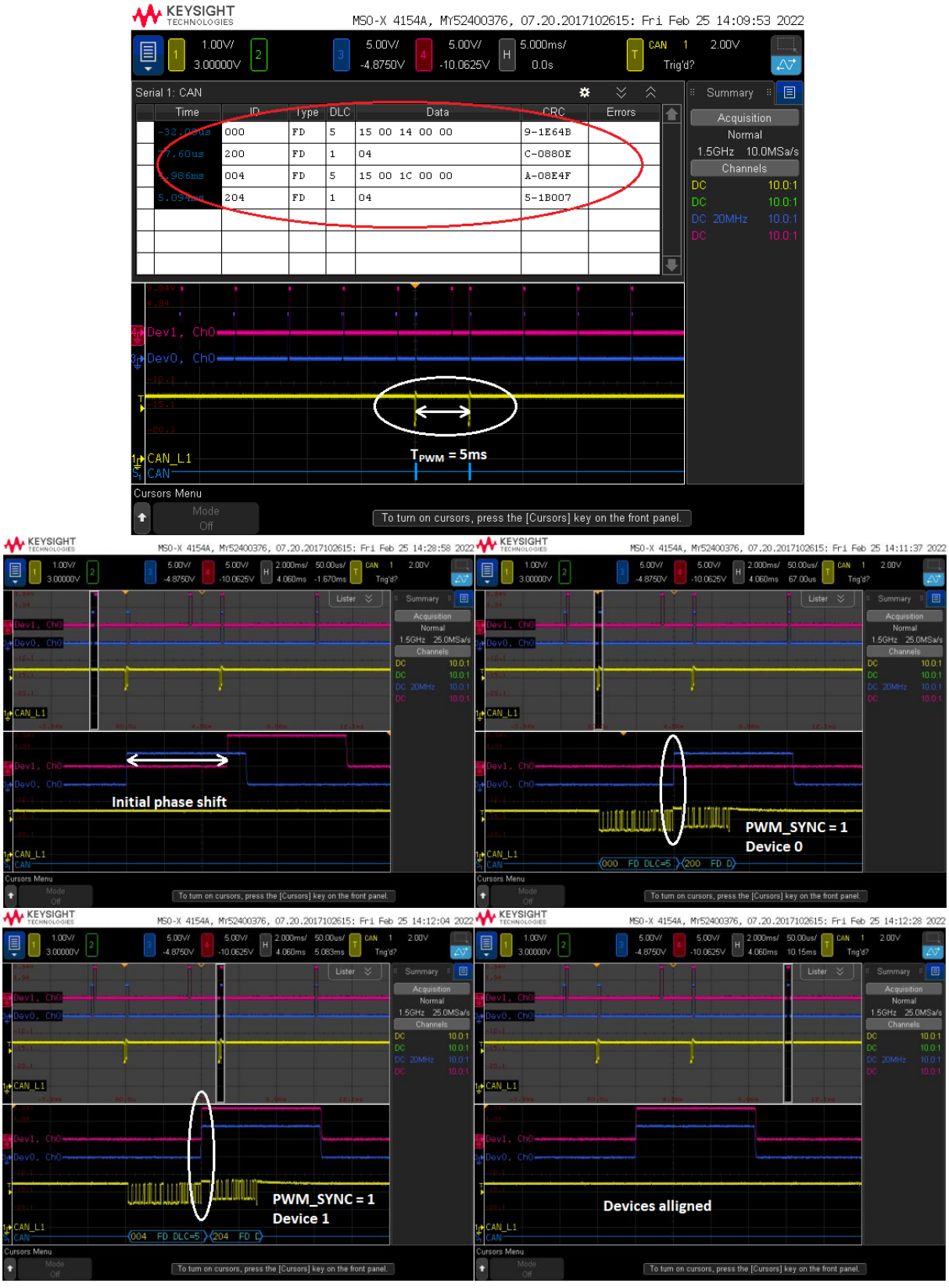
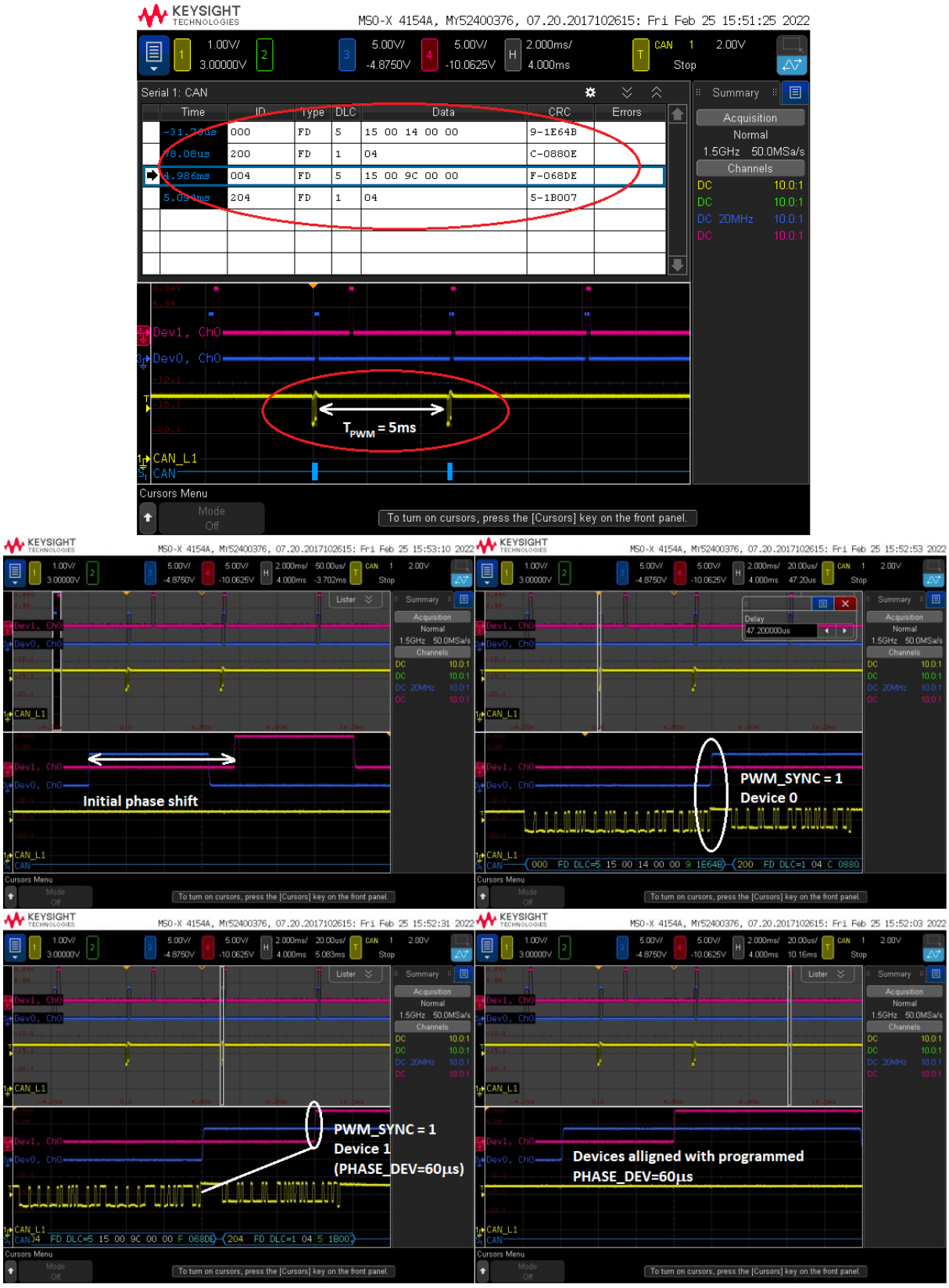


Figure 12. Demonstration of example 2) – PHASE_DEV @ device1 = 60 µs



Appendix A Reference documents

Table 2. Reference documents

Document name	Title
DS12879	32-channel LED driver with automotive CAN FD Light interface- Datasheet

Revision history

Table 3. **Document revision history**

Date	Version	Changes
10-Jun-2022	1	Initial release.

Contents

List of figures