

# AP32101

## TriCore AUDO-NG

Serial Output Expansion using  
HCT595 Shift Register(s) via the  
MSC on AUDO-NG derivatives

Microcontrollers



Never stop thinking.

---

**TriCore AUDO-NG****Revision History:            2005-09**

V 1.0

Previous Version:            -

Page	Subjects (major changes since last revision)

**We Listen to Your Comments**

Any information within this document that you feel is wrong, unclear or missing at all?  
Your feedback will help us to continuously improve the quality of this document.

Please send your proposal (including a reference to this document) to:

**[mcdocu.comments@infineon.com](mailto:mcdocu.comments@infineon.com)**



<b>Table of Contents</b>		<b>Page</b>
<b>1</b>	<b>Introduction</b> .....	<b>4</b>
1.1	Introduction to the Micro Second Channel bus (MSC) .....	4
1.2	HCT595 Information .....	5
<b>2</b>	<b>Example Implementation using the TC1796 StarterKit</b> .....	<b>6</b>
2.1	MSC Configuration .....	7
2.2	Transmission Modes .....	7
2.2.1	Configuration Caveats .....	7
<b>3</b>	<b>Software Implementation</b> .....	<b>9</b>
3.1	Basic configuration for the TC1796 TriBoard .....	9
3.2	Triggered Mode .....	10
3.3	Data Repetition Mode .....	11
3.3.1	GPTA PWM Configuration .....	11
3.3.2	Data Transfer Rate .....	12
3.4	Latency .....	12

# 1 Introduction

This application note provides information to add general purpose or timed outputs using one of the new serial ports available on the AUDO-NG devices. This is easily accomplished thru serial expansion of outputs pins using the Micro Second Channel serial bus available on all AUDO-NG derivatives. One such method described here is to use a standard logic device “54/74HCT595” These are “Serial-to-Parallel” latches that can be easily expanded into multiple groups of eight outputs (per device). A practical example will be described using the TC1796 which is one device from the AUDO-NG family.

## 1.1 Introduction to the Micro Second Channel bus (MSC)

The MSC is a serial interface that is especially designed to connect external power devices to the TC1796. The serial data transmission capability minimizes the number of pins required to connect such external power devices. Parallel data information (coming from the timer units) or command information is sent out to the power device via a high speed synchronous serial data stream (downstream channel). The MSC receives data and status back from the power device via a low-speed asynchronous serial data stream (upstream channel). This implementation only uses the downstream channel as the HCT595 is an Output Only Driver.

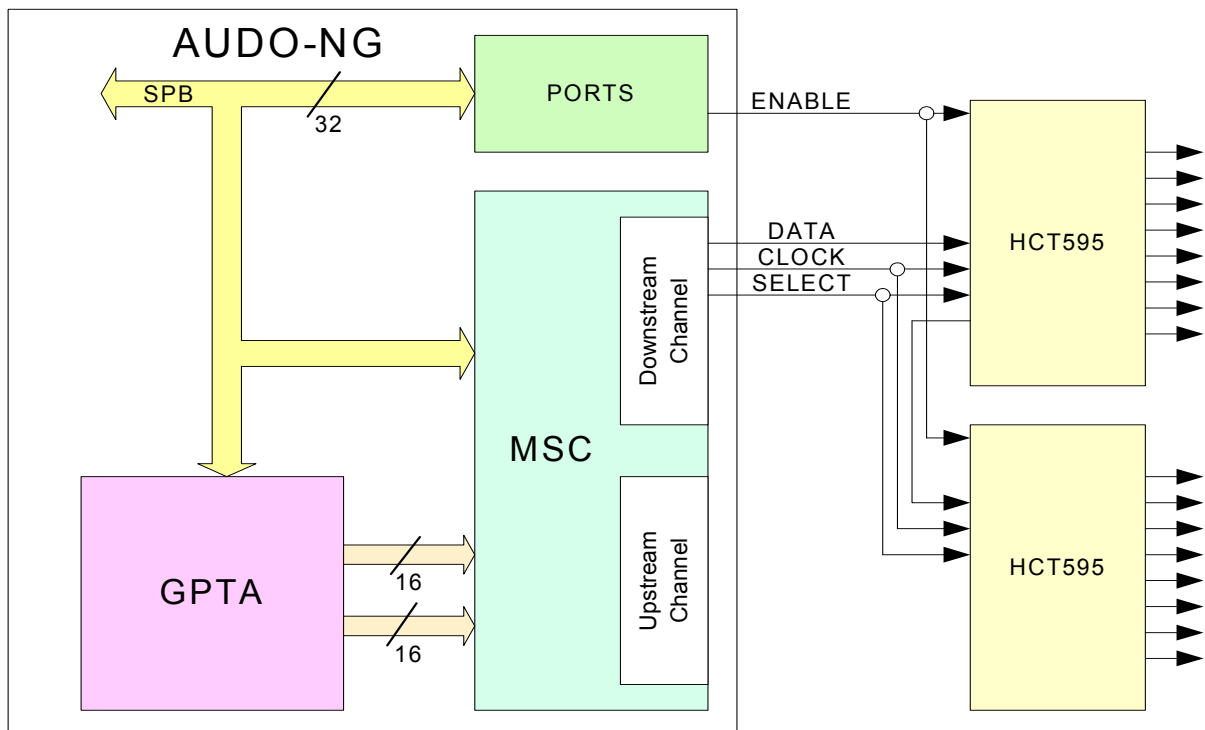


Figure 1 HCT595 connected to AUDO-NG

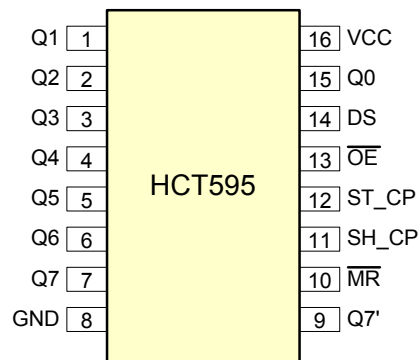
## 1.2 HCT595 Information

The HCT595 is an 8-stage serial shift register with a storage register and 3-state outputs. The shift register and storage register have separate clocks.

Data is shifted on the positive-going transitions of the SH\_CP input and is expected with the most significant bit first (MSB). The data in each register is transferred to the storage register on positive-going transition of the ST\_CP input.

The shift register has a serial input (DS) and a serial output ( $\overline{Q7}$ ) for cascading. There is also an asynchronous reset ( $\overline{MR}$ ) for all of the shift register stages. The data in the storage register appears at the output whenever the output enable input ( $\overline{OE}$ ) is low.

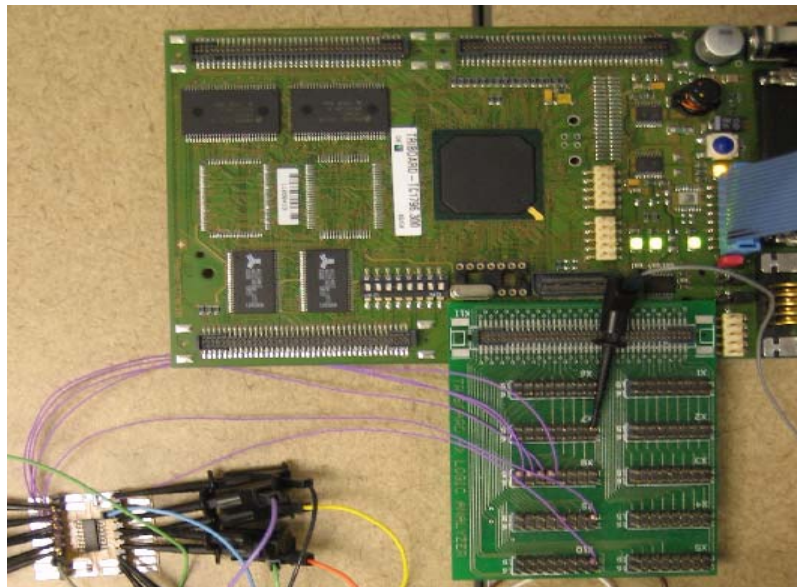
The bus driver outputs have a maximum output source or sink current of 35 mA. The serial clock has a maximum frequency of 100MHz.



**Figure 2 HCT595 Pin Configuration**

## 2 Example Implementation using the TC1796 StarterKit

In this example a small SOIC board populated with a HCT595 was connected to the I/O expansion board of a TC1796 Starterkit. The Micro Second Channel Zero (MSC0) serial peripheral was used to demonstrate the basic concept.



**Figure 3 Hardware Setup**

*Note: Infineon Application Note AP32013 provides another example of using the MSC.*

**Table 1 Pin Interconnect scheme between the HC595 and TC1796**

HC595		TC1796		
Pin	Symbol	Pin	Symbol	Description
1	Q1			Parallel data output
2	Q2			Parallel data output
3	Q3			Parallel data output
4	Q4			Parallel data output
5	Q5			Parallel data output
6	Q6			Parallel data output
7	Q7			Parallel data output
8	GND		GND	Ground
9	Q7'			Connect to DS of the next HCT595 in the chain
10	$\overline{\text{MR}}$	A23	$\overline{\text{HDRST}}$	Reset Indication Output

**Table 1 Pin Interconnect scheme between the HC595 and TC1796**

HC595		TC1796		
Pin	Symbol	Pin	Symbol	Description
11	SH_CP	C20	P9.8	FCLP0B MSC0 Clock Output
12	ST_CP	C19	P9.6	EN01 MSC0 Device Select Output 1
13	$\overline{OE}$	D19	P9.5	GPIO (Used to enable data output drivers)
14	DS	D20	P9.7	SOP0B MSC0 Serial Data Output
15	Q0			Parallel data output
16	VCC			+5V

## 2.1 MSC Configuration

For a detailed understanding of the MSC the user is advised to read the TC1796 Peripheral User’s Manual. This application note will only describe the needed information to make the proper configuration and usage with the HCT595 and MSC0.

*Note: The TC1796 allows for a maximum expansion of 56 outputs. 32 outputs on MSC0 and 24 outputs on MSC1.*

## 2.2 Transmission Modes

There two transmission modes available for sending downstream data, [triggered mode](#) and [data repetition mode](#).

To send a downstream data frame in triggered mode the user must first write the data to be sent into the Data Downstream register. Then set the bit SDP in the ISC register which causes the data to transmitted when the bus is found to idle.

In Data Repetition Mode, data frames are sent out continuously without any software interaction. In the time gap between two consecutive data frames, passive time frames can be inserted. When expanding timed outputs from the GPTA, the user should select the Data high speed Mode for autonomous transmission.

### 2.2.1 Configuration Caveats

Q) How is the data shifted out of the MSC, MSB or LSB?

A) The MSC will present the LSB of the data to be shifted out first. However the HCT595 expects the data to be shifted MSB first. **This should not present a problem as long as the user understands this and correctly maps the functionality of the pins to the data being sent** (see [Figure 4](#)).

Q) What are the phase and mode requirements for the serial transfer?

A) There is only the possibility to change the polarity of the clock and data signals. By inverting the polarity of the clock we are able to run in SPI mode 0.

The test platform used in this application only has one HCT595 connected to the TC1796. Therefore the settings are given for this particular setup. One data byte is used to send output data controlled from software. Thus software can control bits 1 thru 7 of the data byte (Q0..Q6). Bit 0 (Q7) is directly controlled from the GPTA and its selection is shown by the ALTINL configuration.

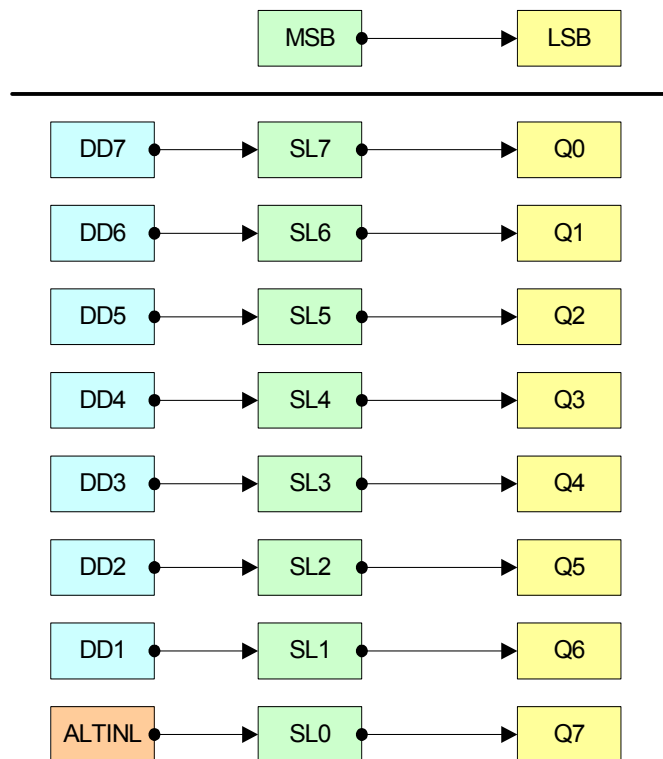


Figure 4 Relationship between the MSC data register bits and the outputs of the HCT595



### 3 Software Implementation

The development suite for this example was made with Tasking V2.2r3. All of the source code and project file have been included in the corresponding zip file.

#### 3.1 Basic configuration for the TC1796 TriBoard

The following software settings are programmed into the MSC0 of the TC1796. The user is free to modify the clock/frequency settings. The settings given in this example are purely arbitrary.

- 8 MHz crystal on the TQ-Components StarterKit
- PLL configured to run at 60 MHz
- Watchdog has been disabled
- GPTA0 is configured to run at 20MHz
  - 5 LTC cells configured for coherent update of Duty cycle and Period
  - Uses cells LTC24..LTC28
  - Output on P4.0
  - Output on MSC0 bit 0 which is Q7 of the HCT595
- MSC0 clock is configured for 20 MHz
  - MSC0 baud rate for data transmission is 10 MHz.
  - Two mode options (user configurable)
    - Triggered Mode
    - Data Repetition Mode

Main software loop...

```
while(1) {
    /* indicate activity */
    if (counter++ > 10000u) {
        counter = clear;
    }
    /* software update */
    if (update == 1) {
        MSC0_DD.U = data;
        MSC0_ISC.B.SDP = true;
        update = clear;
    }
    /* turn off repeat mode */
    if (update == 2) {
        MSC0_DSC.B.TM = false;
        update = clear;
    }
    /* turn on repeat mode */
    if (update == 3) {
        MSC0_DSC.B.TM = true;
        update = clear;
    }
    /* Change duty cycle */
    if (update == 4) {
```

```

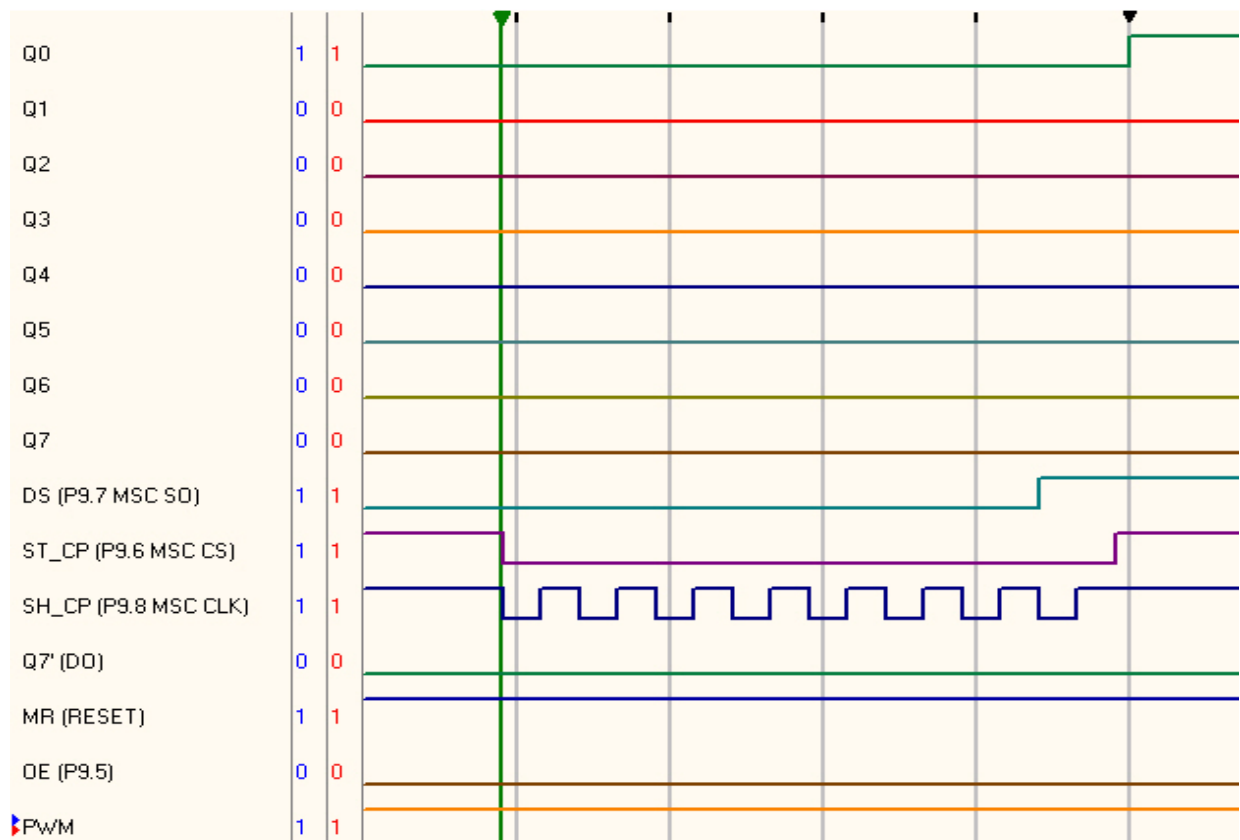
ChangeDutyCycle(dutyCycle);
update = clear;
}
/* Change period */
if (update == 5) {
ChangePeriod(period);
update = clear;
}
/* Change period and duty cycle */
if (update == 6) {
ChangePeriodDutyCycle(period, dutyCycle);
update = clear;
}
};

```

### 3.2 Triggered Mode

To update the data that is being sent on the MSC0 manually, simply write a new value to the Data variable. Then set the variable “Update” equal to 1. Now, when the loop is executed the MSC0 will be updated as shown in [Figure 5](#).

*Note: The variable data is equal to 0x80.*



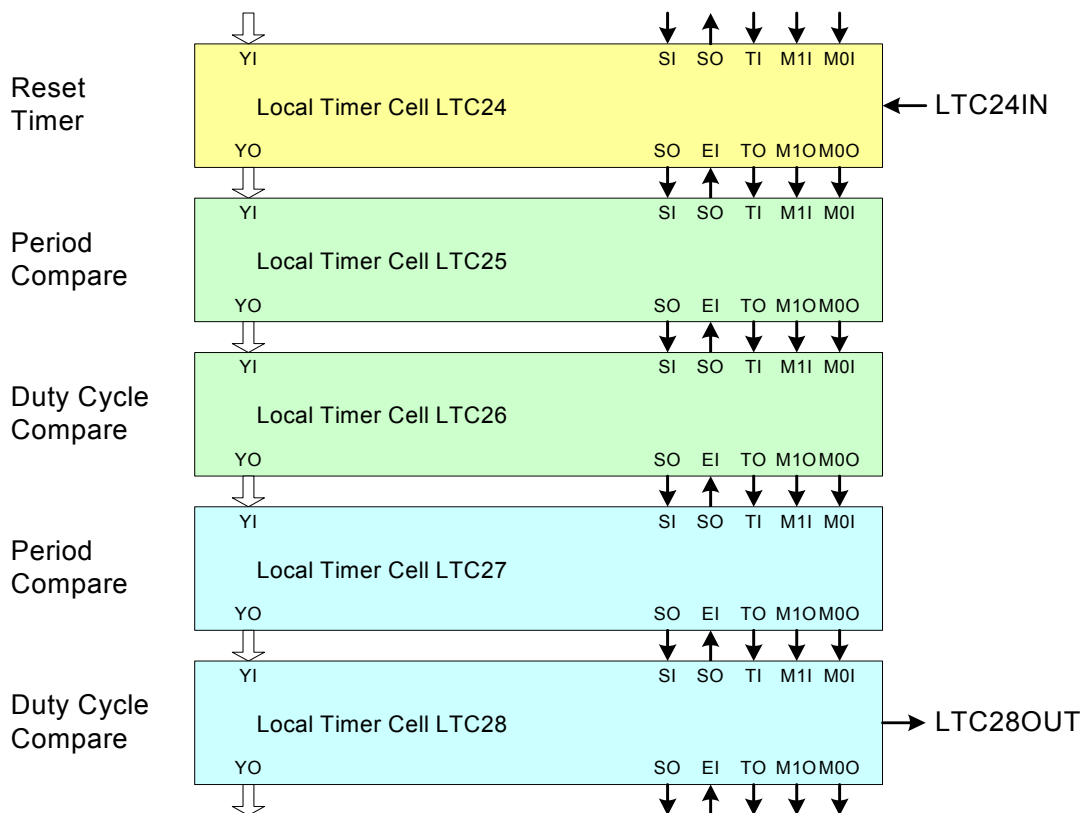
**Figure 5 Single Transfer Software Update**

### 3.3 Data Repetition Mode

When the transmission mode of the MSC0 is configured for Data Repetition Mode the packet of data is transmitted at a periodic rate. This configuration would typically be used when the source data contains asynchronous changes from the GPTAx peripheral.

#### 3.3.1 GPTA PWM Configuration

As shown in **Figure 6**, a logical unit of five LTCs can be used to generate a PWM signal with a programmable duty cycle, period length, and fully coherent update of the period and duty cycle. In this example, LTC24 up to LTC28 are used to produce a PWM signal at the output of LTC28 and ALTINL.



**Figure 6 PWM Signal Generation with LTCs (Full Coherent Update)**

In this example the GPTA0 has been configured with one PWM that is output on Port 4.0 and also on bit 0 of the MSC0 (Q7 of the HCT595). The PWM configuration uses 5 LTC's to support full coherent updating of the Period or Duty Cycle.

### 3.3.2 Data Transfer Rate

A data transfer can consist of two 16-bit words, high and low select bits along with a programmable period delay between the transfers.

$$\text{Number of cycles} = \text{ENSELL} + \text{NDBL} + \text{ENSELH} + \text{NDBH} + \text{PPD} \quad [1]$$

$$\text{Number of cycles} = \text{ENSELL} + \text{NDBL} + \text{PPD} \quad [2]$$

In this example, only 8-bits are being transmitted so we will use [Equation \[2\]](#).

Where:

ENSELL = 0/1 (bit is used in the calculated whether sent or not)

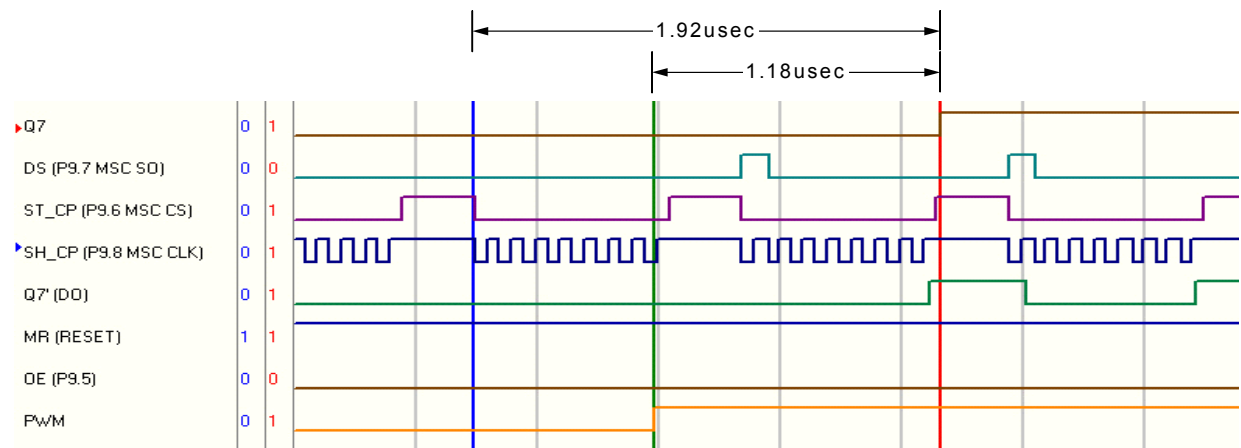
NDBL = 8

PPD = 2

which results in a transfer packet of 11 bit times (1.1usec @ 10MHz clock).

### 3.3.3 Latency

Since the data is serialized the user should carefully consider the data transfer rate and choose only to serialize timed I/O that can tolerate this added latency of the signals. The worst case latency can be calculated by 2 data transfers on the MSC0.



**Figure 7**      **GPTA PWM Change reflected in the MSC0 output**

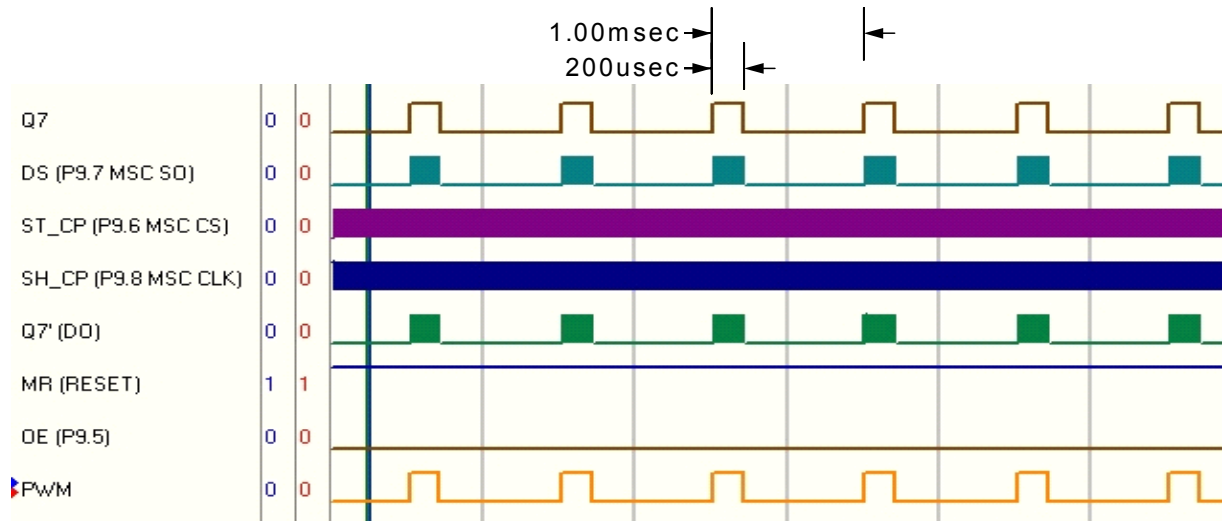


Figure 8 PWM Overview on GPTA pin and HCT595 pin via MSC0

<http://www.infineon.com>

Published by Infineon Technologies AG