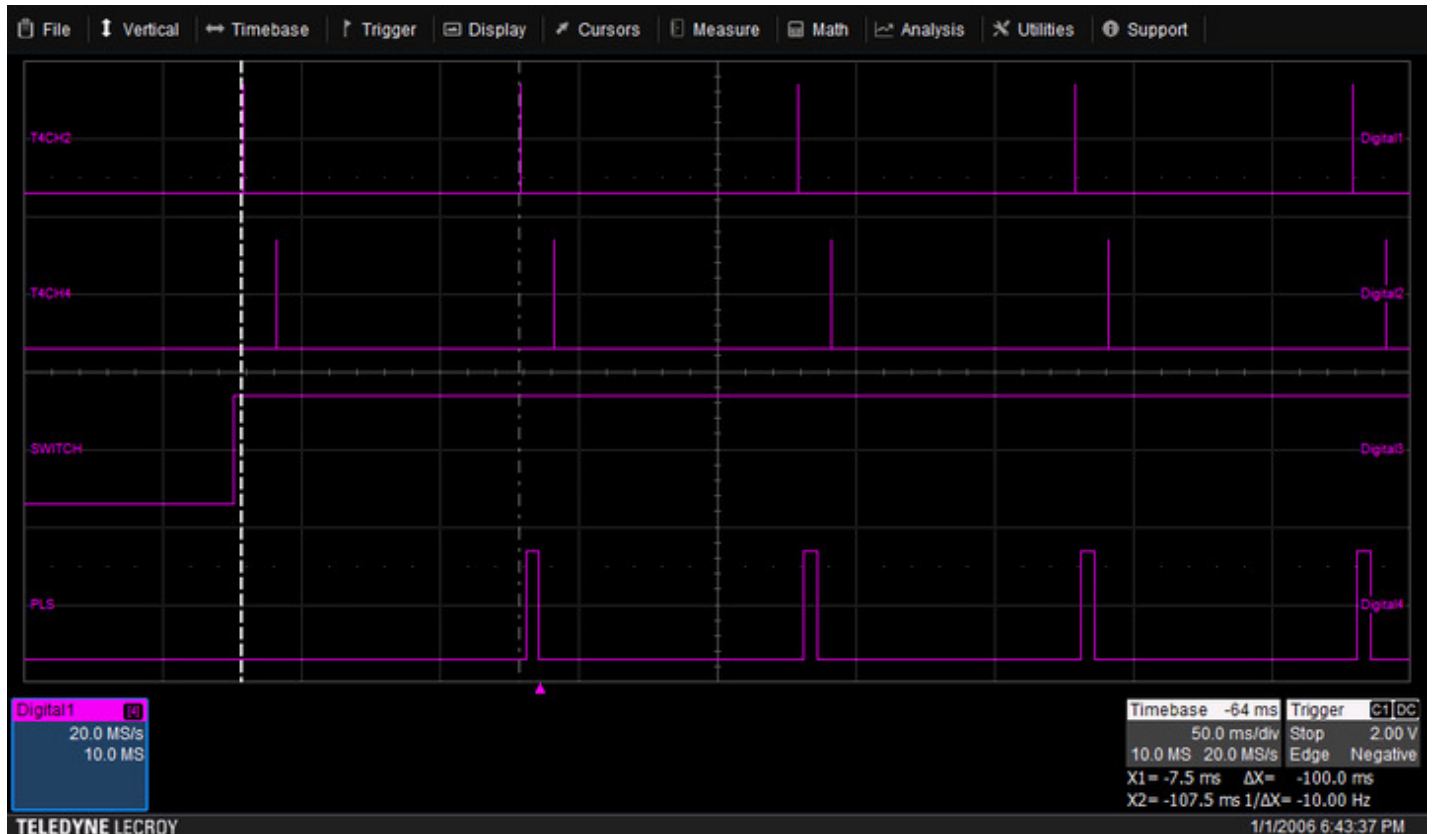


Hi Guys,

I have some issues with my master-slave time configuration. I simply chain TIM4->TIM1, whereas TIM4 triggers TIM1 every Xxx ms. As a result TIM1 responds with a single pulse on channel 1.

The issue is now that the first pulse of TIM1 is not output as you can see in the picture below. You can also see that the master trigger (channel 2) is generated, but not the trigger for the slave-timer.

Those somebody have a hint for me? Or can point me into the right direction?



Here the according code:

```

/* === variables === */
/* Master */
TIM_HandleTypeDef      TimMasterHandle  = {0};
TIM_MasterConfigTypeDef sMasterConfig  = {0};
TIM_OC_InitTypeDef     sMasterOCConfig  = {0};

/* Slave */
TIM_HandleTypeDef      TimSlaveHandle   = {0};
TIM_SlaveConfigTypeDef sSlaveConfig    = {0};
TIM_OC_InitTypeDef     sSlaveOCConfig   = {0};

#ifdef DEBUG_MASTER
/* === code === */
__TIM4_CLK_ENABLE();

TimMasterHandle.Instance = TIM4;

/* default Master configuration: TIM4 */
TimMasterHandle.Init.ClockDivision = 0;
TimMasterHandle.Init.Prescaler     = 139;
TimMasterHandle.Init.RepetitionCounter = 0;
TimMasterHandle.Init.CounterMode   = TIM_COUNTERMODE_UP;

```

```

TimMasterHandle.Init.Period          = 60000;

if(HAL_TIM_PWM_Init(&TimMasterHandle) != HAL_OK ){

    /* error */
}

/* Configure the PWM_channel_2 */
sMasterOCConfig.OCMode              = TIM_OCMODE_PWM1;
sMasterOCConfig.OCpolarity          = TIM_OCPOлярITY_HIGH;
sMasterOCConfig.OCNPolarity        = TIM_OCNPOLARITY_HIGH;
sMasterOCConfig.OCFastMode         = TIM_OCFAST_ENABLE;
sMasterOCConfig.OCIdleState        = TIM_OCIDLESTATE_SET;
sMasterOCConfig.OCNIdleState       = TIM_OCNIDLESTATE_SET;
sMasterOCConfig.Pulse              = 2000;

if(HAL_TIM_PWM_ConfigChannel(&TimMasterHandle, &sMasterOCConfig, TIM_CHANNEL_2) != HAL_OK)
{
    /* Configuration Error */
}

sMasterOCConfig.Pulse              = 9000;
if(HAL_TIM_PWM_ConfigChannel(&TimMasterHandle, &sMasterOCConfig, TIM_CHANNEL_4) != HAL_OK)
{
    /* Configuration Error */
}

/* Configure TIM2 as master & use the update event as Trigger Output (TRGO) */
sMasterConfig.MasterOutputTrigger = TIM_TRGO_OC2REF;
sMasterConfig.MasterSlaveMode     = TIM_MASTERSLAVEMODE_ENABLE;
if( HAL_TIMEx_MasterConfigSynchronization(&TimMasterHandle,&sMasterConfig) != HAL_OK)
{
    /* Configuration Error */
}
#endif

#ifdef DEBUG_SLAVE
/* === code === */
__TIM1_CLK_ENABLE();

/* Set TIMx instance */
TimSlaveHandle.Instance = TIM1;

TimSlaveHandle.Instance = TIM1;

/* default Master configuration: TIM4 */
TimSlaveHandle.Init.ClockDivision = 0;
TimSlaveHandle.Init.Prescaler     = 27;
TimSlaveHandle.Init.RepetitionCounter = 0;
TimSlaveHandle.Init.CounterMode   = TIM_COUNTERMODE_UP;
TimSlaveHandle.Init.Period        = 60000;

if(HAL_TIM_PWM_Init(&TimSlaveHandle) != HAL_OK ){

    /* error */
}

/* Configure the PWM_channel_2 */
sSlaveOCConfig.OCMode              = TIM_OCMODE_PWM1;

```

```
sSlaveOCConfig.OCpolarity      = TIM_OCPOLARITY_HIGH;
sSlaveOCConfig.OCNPolarity     = TIM_OCNPOLARITY_HIGH;
sSlaveOCConfig.OCFastMode      = TIM_OCFAST_ENABLE;
sSlaveOCConfig.OCIdleState     = TIM_OCIDLESTATE_RESET;
sSlaveOCConfig.OCNIdleState    = TIM_OCNIDLESTATE_RESET;
sSlaveOCConfig.Pulse           = 30000;

if(HAL_TIM_PWM_ConfigChannel(&TimSlaveHandle, &sSlaveOCConfig, TIM_CHANNEL_1) != HAL_OK)
{
    /* Configuration Error */
}

/* Configure TIM3 in Gated slave mode &
use the Internal Trigger 0 (ITR0) as trigger source */
sSlaveConfig.SlaveMode        = TIM_SLAVEMODE_TRIGGER;
sSlaveConfig.InputTrigger     = TIM_TS_ITR3;
if(HAL_TIM_SlaveConfigSynchronization(&TimSlaveHandle, &sSlaveConfig) != HAL_OK)
{
    /* Configuration Error */
}

HAL_TIM_OC_ConfigChannel(&TimSlaveHandle, &sSlaveOCConfig, TIM_CHANNEL_1);

SET_BIT(TIM1->CCER, TIM_CCER_CC1E);
SET_BIT(TIM1->BDTR, TIM_BDTR_MOE);

WRITE_REG(TIM1->SR, 0);

/* Enable the TIM Capture/Compare 2 interrupt */
__HAL_TIM_ENABLE_IT(&TimSlaveHandle, TIM_IT_CC1);

/* Set timer interrupt to the lowest priority */
HAL_NVIC_SetPriority(TIM1_CC_IRQn, 15, 0); /* GLOBAL FILE FOR PRIORITY SETTINGS ??? */

/* Enable capture and compare interrupt */
HAL_NVIC_EnableIRQ(TIM1_CC_IRQn);

HAL_NVIC_ClearPendingIRQ(TIM1_CC_IRQn);

/* Set timer interrupt to the lowest priority */
HAL_NVIC_SetPriority(TIM4_IRQn, 15, 0); /* GLOBAL FILE FOR PRIORITY SETTINGS ??? */

/* Enable capture and compare interrupt */
HAL_NVIC_EnableIRQ(TIM4_IRQn);

HAL_NVIC_ClearPendingIRQ(TIM4_IRQn);

/* Enable the TIM Capture/Compare 2 interrupt */
__HAL_TIM_ENABLE_IT(&TimMasterHandle, TIM_IT_CC2);

/* Enable the TIM Capture/Compare 4 interrupt */
__HAL_TIM_ENABLE_IT(&TimMasterHandle, TIM_IT_CC4);
```