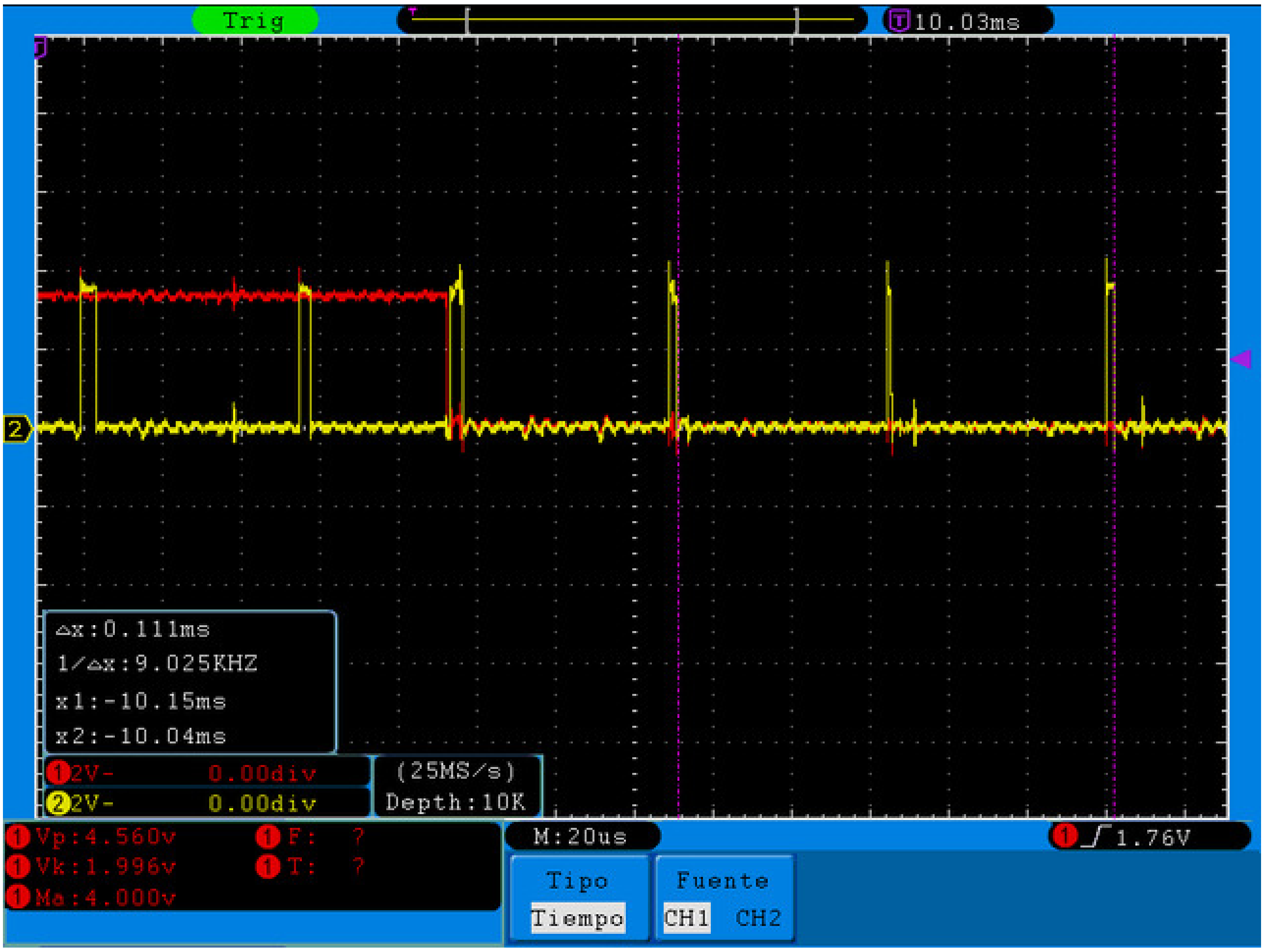
I am doing a program where I have a 50Hz square signal and a PWM to generate a sinusoidal with 90 points. Each time the signal edge of 50Hz is raised or lowered, the timer must be reset and start a new modulation. However, the first two values do not match the programmed width.



The modulation starts correctly on the third pulse, and when there is this displacement the modulation does not end correctly. This is the code:

This is the external interrupt, for sync my 50Hz signal.

```
void EXTI15_10_IRQHandler(void)
{
//PB10 50H+          input float int
//PB11 50H-          input float int
     if(EXTI_GetITStatus(EXTI_Line10) != RESET)
     {
          if(MAR==1){
               //port_set(PORT_B, 15, 1); //MAR ON
               port_set(PORT_A, 4, 1);  //ON 50Hz
               //TIM2->CCR1=0;
               portsONtimer();
          }
    EXTI_ClearITPendingBit(EXTI_Line10);
     }
          if(EXTI_GetITStatus(EXTI_Line11) != RESET)
     {
          if(MAR==1){
               port_set(PORT_A, 4, 0);  //OFF 50Hz
               portsONtimer();
               }
          //port_set(PORT_A, 4, 0);       //OFF 50Hz
    EXTI_ClearITPendingBit(EXTI_Line11);
```

```
     }
}????????????????????????
```

This is the timer controlling the PWM:

```
void TIM2_IRQHandler(void)
{
     //     static int indice = 0;
          TIM2->SR &= ~(1<<0);  // clear UIF flag               - bit0
          switch(dir){
               case 0:
                    if(indice==89){ //change direction of the table
                         dir= 1;
                    }
                    TIM2->CCR1=frec50[indice]; //Increasing table, 1/4 sine
               break;

               case 1:
                    if(indice==89){ //change direction of the table
                         dir= 0;
                         stop_timer();
                    }
                    TIM2->CCR1=frec50[PWM_ELEMENTS - indice - 1];     //Decreasing table, 2/4 sine

               break;
          }
          indice = (indice + 1) % PWM_ELEMENTS;
}??????????????????????????????????????????????
```

And this is the function that starts timer:

```
void portsONtimer(void){

     //GPIOA->CRL = (GPIOA->CRL & 0xFFFFFFF0 | 0x00000003); //PA0.1
     //port_set(PORT_A, 0, 0); //S
     dir= 0;
     indice=0;
     TIM_Cmd(TIM2, ENABLE);
     //TIM2->CCR1=0;
     TIM2->CNT = 0;
     GPIOA->CRL = (GPIOA->CRL & 0xFFFFFFF0 | 0x0000000B); //PA0
} ???????????
```

And the stop timer:

```
void stop_timer(void){
     TIM2->CCR1=0; //negat
     //TIM1->CCR2=0;
     TIM_Cmd(TIM2, DISABLE);
     port_set(PORT_A, 0, 0); //S
     GPIOA->CRL = (GPIOA->CRL & 0xFFFFFFF0 | 0x00000003); //PA0.1
     TIM2->CNT = 0;
     indice=0;
     dir= 0;
     TIM2->CCR1=0;
}
???????????????????????
```

Any idea?
Thank you very much.

Daniel Prieto