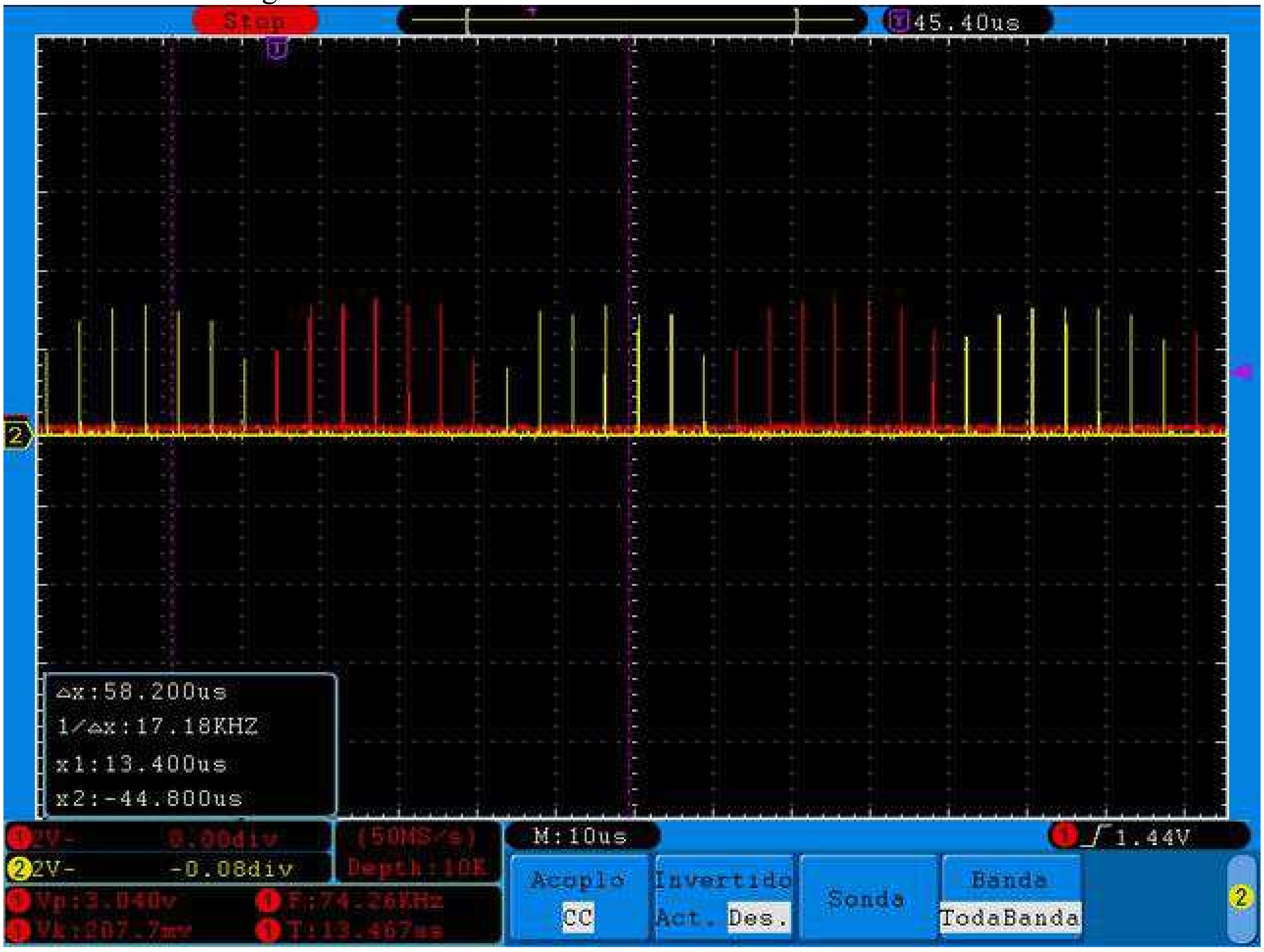
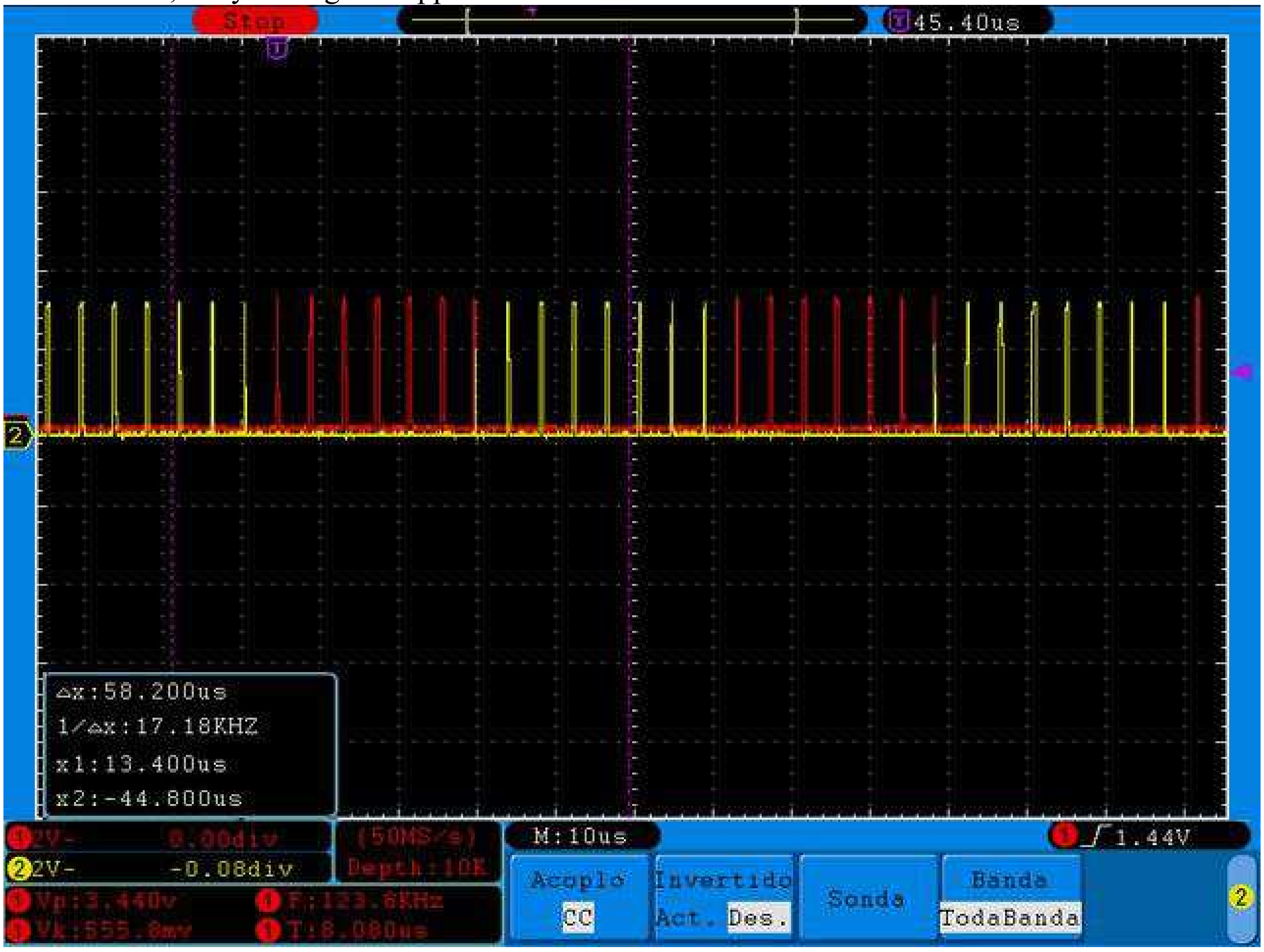
Hi, I'm using the PWM to create a sinusoid at a desired frequency and amplitude.
I have a table with the values, and I am updating it during the execution of the program.
But I have an unexpected glitch that overlaps with the other modulation, and that would make me drive both transistors at the same time ... a total ruin in my application.

Where does this glitch come from?
I put some captures and my code:

Modulation without glitch:



After a while, the yellow glitch appears in the left:



```
void DMA_Configuration(void)
{
 DMA_InitTypeDef DMA_InitStructure;

 DMA_Cmd(DMA1_Channel3, DISABLE);
 /* DMA1 Channel5 Config */
 DMA_DeInit(DMA1_Channel3);
DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)TIM3_CCR2_Address; //TIM1_CCR3_Address;
 DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)&tim3B; //SRC_Buffer;
 DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralDST;
 DMA_InitStructure.DMA_BufferSize = 14;
 DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
 DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
 DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
 DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
 DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
 DMA_InitStructure.DMA_Priority = DMA_Priority_High;
 DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel3, &DMA_InitStructure);
/* DMA1 Channel3 enable */

 DMA_Cmd(DMA1_Channel3, ENABLE);

 DMA_Cmd(DMA1_Channel2, DISABLE);
 DMA_DeInit(DMA1_Channel2);
DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)TIM3_CCR1_Address;
 DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)&tim3A; //SRC_Buffer;
 DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralDST;
 DMA_InitStructure.DMA_BufferSize = 14;
 DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
 DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
 DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
 DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
 DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
 DMA_InitStructure.DMA_Priority = DMA_Priority_High;
 DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel2, &DMA_InitStructure);
 /* DMA1 Channel2 enable */


 DMA_Cmd(DMA1_Channel2, ENABLE);

}???????????????????????????????????????????

//Tables with initial values
u16 tim3A[14]={114,211,276,299,276,211,114,0,0,0,0,0,0,0};
u16 tim3B[14]={0,0,0,0,0,0,0,114,211,276,299,276,211,114};???

void TIM3_config(void){

     u16 time;

     DMA_Configuration(); //DMA1_chanel5

     RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);
     RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);

     GPIOA->CRL = (GPIOA->CRL & 0x00FFFFFF | 0xBB000000); //PA6-7

     time = ((maxfrec * midfrec)/256); //35kHz 256

  TIM_TimeBaseStructure.TIM_Prescaler = 0;
  TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
  TIM_TimeBaseStructure.TIM_Period = time;
  TIM_TimeBaseStructure.TIM_ClockDivision = 0x0;
  TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);

 // TIM_TimeBaseStructure.TIM_ClockDivision = 0x0;
  TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);

  /* Channel 1 Configuration in PWM mode */
  TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM2;
  TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
  TIM_OCInitStructure.TIM_OutputNState = TIM_OutputNState_Disable;
  TIM_OCInitStructure.TIM_Pulse = 0;
  TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_Low; //Low
  TIM_OCInitStructure.TIM_OCNPolarity = TIM_OCNPolarity_Low;
  TIM_OCInitStructure.TIM_OCIdleState = TIM_OCIdleState_Set;
  TIM_OCInitStructure.TIM_OCNIdleState = TIM_OCIdleState_Reset;

  TIM_OC1Init(TIM3, &TIM_OCInitStructure);
  TIM_OC2Init(TIM3, &TIM_OCInitStructure);


     /* TIM3 Update DMA Request enable */
  TIM_DMACmd(TIM3, TIM_DMA_Update, ENABLE);

     TIM_DMACmd(TIM2, TIM_DMA_Update, ENABLE);

//     TIM_DMACmd(TIM3, TIM_DMA_CC2, ENABLE);

     TIM_Cmd(TIM2, ENABLE);
     TIM_Cmd(TIM3, ENABLE);
     TIM_CtrlPWMOutputs(TIM3, ENABLE);

}


void newtimereal(void){
     u8 i;

     for(i=0;i<4;i++){
          timreal[i] = ((timmax[i] *modul256 *modfrec256)/256)/512;
     }
```

```
}

//change frec different
void calculatetimereal(void){

     newtimereal();

     //CHARGE NEW FRAME *2

     tim3A[0]=timreal[0];
     tim3A[1]=timreal[1];
     tim3A[2]=timreal[2];
     tim3A[3]=timreal[3];
     tim3A[4]=timreal[2];
     tim3A[5]=timreal[1];
     tim3A[6]=timreal[0];

     //tim3B[6]=0;
     tim3B[6]=timreal[0];
     tim3B[7]=timreal[1];
     tim3B[8]=timreal[2];
     tim3B[9]=timreal[3];
     tim3B[10]=timreal[2];
     tim3B[11]=timreal[1];
     tim3B[12]=timreal[0];
     tim3B[13]=0;
}????????????????????????????????????????????????????????????????????????????????????
```

I can post more code if you need.

Thank you in advance!