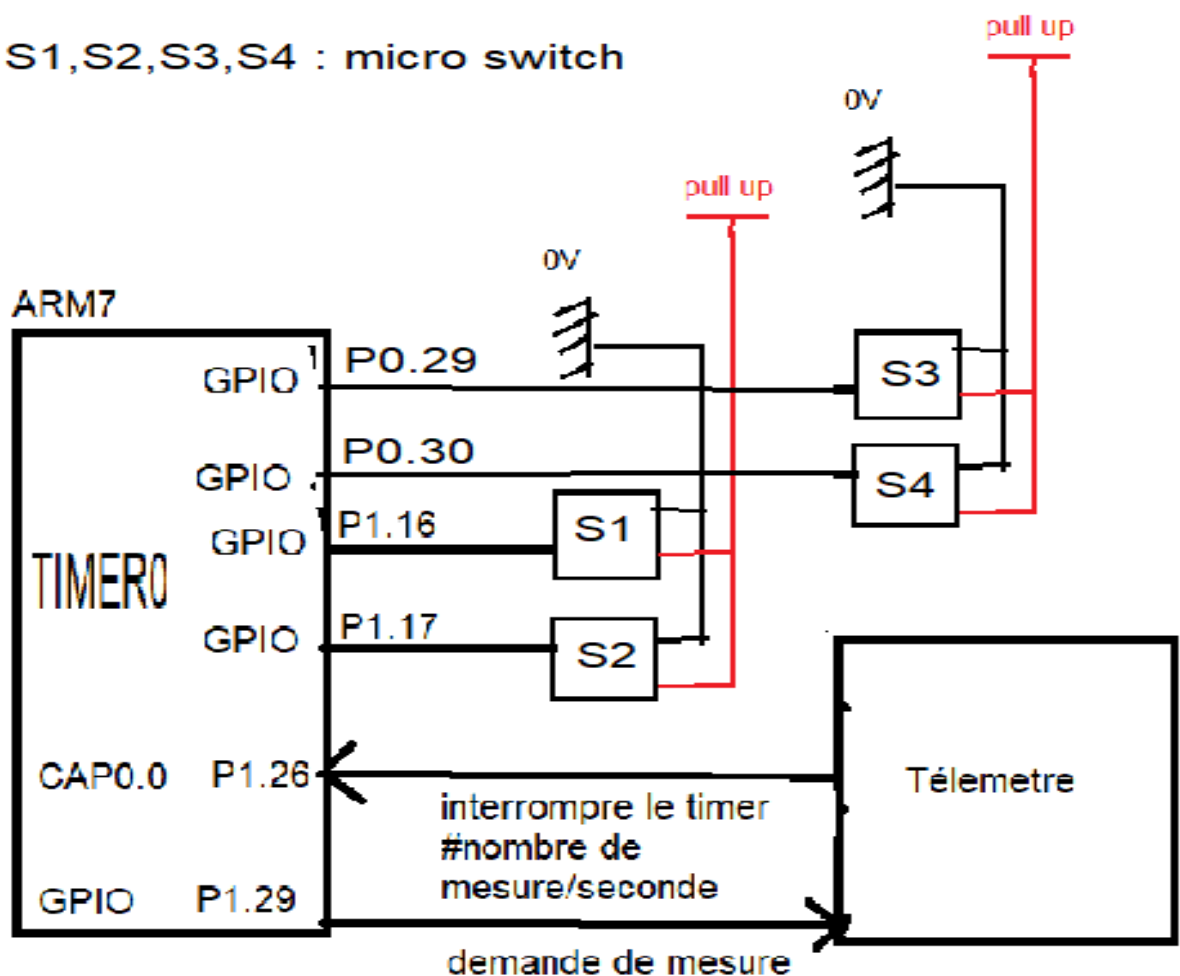


## Fonctionnement du télémètre à ultrasons

**But de module** : Calcul de la distance via les ultrasons entre le robot et l'obstacle

✓ Choix et gestion d'entrées/sorties :

S1,S2,S3,S4 : micro switch

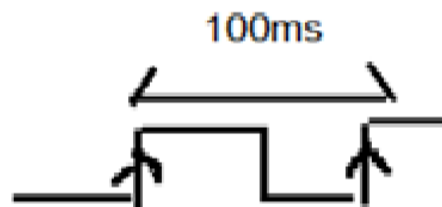


✓ Relation entre le Timer et P1.16 et P1.17 :

P1.17	P1.16	nb/sec	TOMR 1	période	TOMR 0	Rcy %
0	0	10	1200000	100ms	1197600	0.2
0	1	15	800000	66.7ms	797600	0.3
1	0	20	600000	50 ms	597600	0.4
1	1	25	480000	40ms	477600	0.5

1 sec /10 mes =0.1s = 100ms  
 $\Rightarrow f = 10 \text{ hz} \Rightarrow \text{TOMR0} = f_0 / f = 1200000$

la fréquence du ARM7=12 MHZ



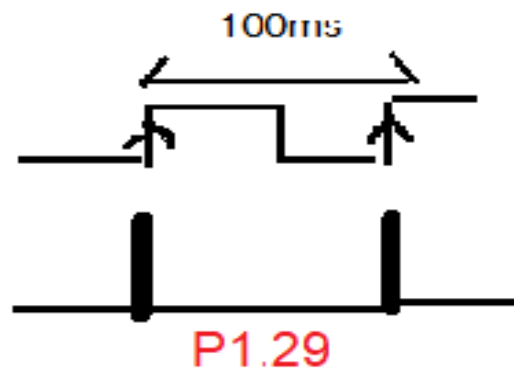
Chaque front montant on a une mesure donc pour  
 10 mesures nous avons 10 fronts montants

Cela veut dire que chaque 100 ms on fait une interruption  
 Donc on cherche la valeur correspondante pour TOMR0

✓ La demande de mesure de distance P1.29 :

- un signal périodique avec un niveau haut 200µs injecté à chaque front montant qui correspond au nombre de mesure exemple pour 10 mes/sec =>

un rapport cyclique :  $200\mu\text{s} / 100\text{ms} = 0.2\%$



réception du signal de l'echo : P1.26  
CAP0.0 : faire une capture au TC( time counter) à chaque détection de front

## PROGRAMATION :

✚ Code C de l'init de Timer :

```
void init_timer0(){  
    //---timer 0  
    T0MCR = 0x19; //Interruption + Raz;  
    T0TCR = 1; //counter ON  
    VICVectAddr4 = (unsigned long)timer0;  
    VICIntEnable = 1 << 4; //canal TIMER0  
    // initialisation de la capture  
}
```

✚ Code de l'interruption :

```
void timer0(void) __irq{ // interruption Nombres mes/sec  
    etat++; i++;  
    if(i==1) {NB_Av=NB_mesure;} // etat avant de NB_mesure  
    if(i==2) {NB_Ap=NB_mesure;} // etat apres de NB_mesure  
    if(NB_Av>NB_Ap) {etat=1;}  
    if(etat==1) {FIO1PIN ^=1<<29; T0CCR=1; T0IR=1; } // mettre à 1 aussi une capture pour T0TC => T0MCR0  
    if(etat==2) {FIO1PIN ^=1<<29; T0IR = 2; } // remettre à 0  
  
    if(etat==3) etat=0;  
    if(i==3) i=0;  
    VICVectAddr = 0;  
}
```

## Code de main :

```
int main(){

    init_GPIO();
    init_timer0();

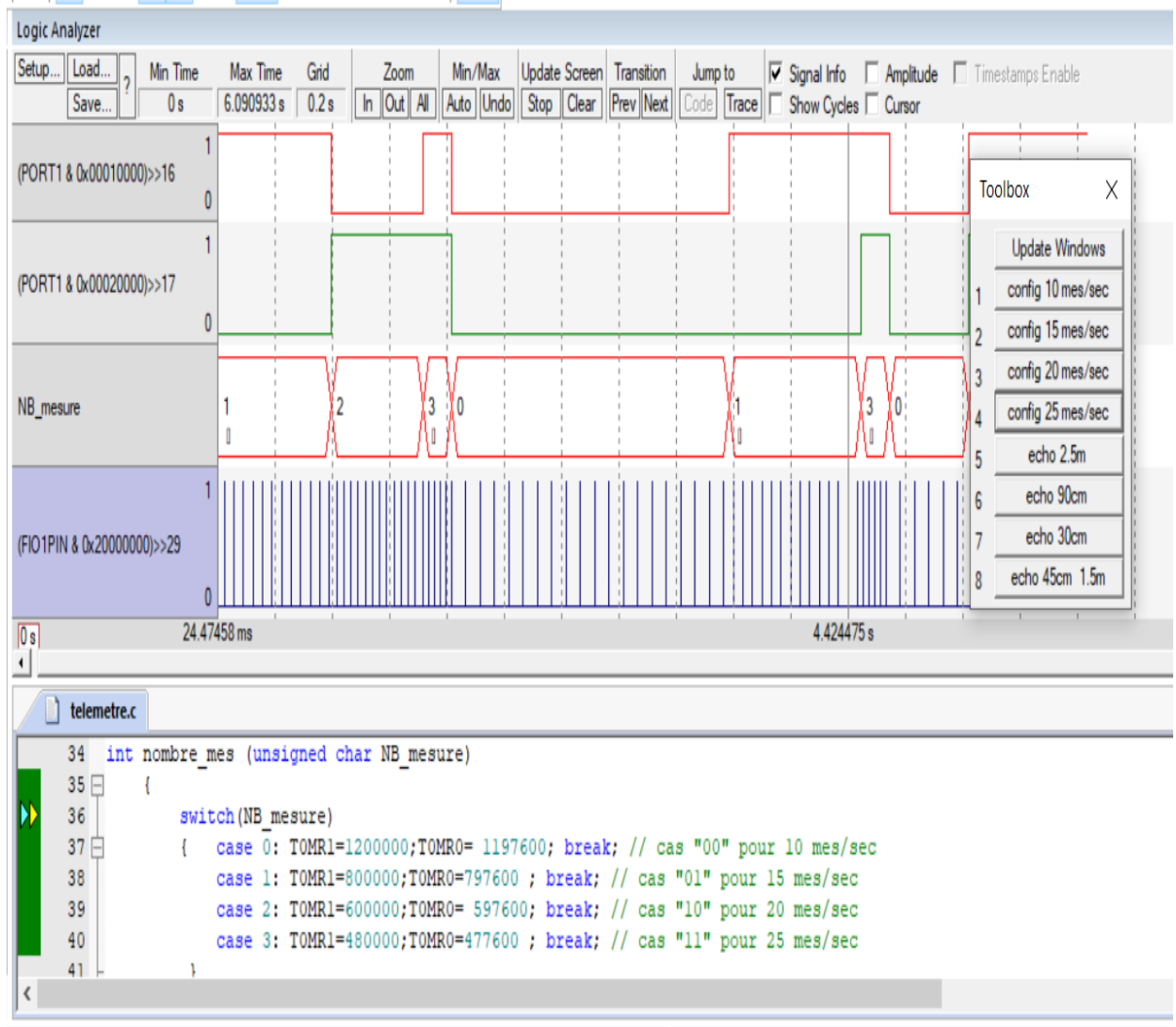
    while(1){
        // selection de Nombre de mesure / sec -----
        NB_mesure= (FIO1PIN & 0x00030000)>>16; // récupérer les P1.16 et P1.17
        nombre_mes (NB_mesure); // récupérer la valeur de T0MR0 selon les boutons appuyés
        //--test de la capture CAP0.0 => P1.26 -----
        T0CCR |=(((FIO1PIN & 0x04000000)>>26) & 0x1)<<3; // ; // => activation T0MRC1 <= T0TC
        if (T0CCR==8) {dt=(T0CR1-T0CR0);distance=(dt/58);} // précision 1 cm et la vitesse du son 345 m/s et la distance en mm

        // mode debug -----
        config_debug=(FIO0PIN & 0x60000000)>>29; // récupérer les P0.29 et P0.30
        debug_telem (config_debug);

    };
}
```

## Chronogrammes :

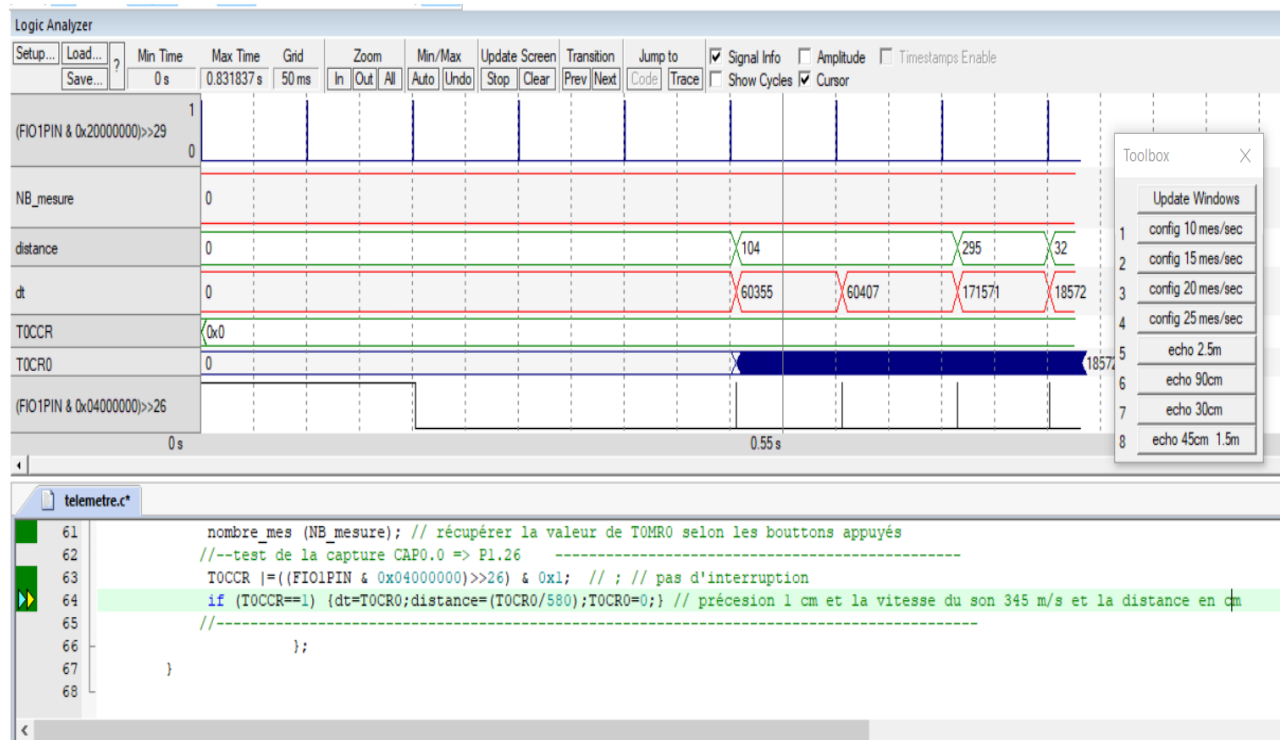
### Simulation de nombre de mesure par seconde



On remarque bien que le nombre de mesures par seconde s'adapte quand on change la configuration nb mes/sec.

## Simulation de la capture :

Utilisation des registres **Count Control Registe T0CCR** et **Timer Control Registe T0CR\_**



On remarque que la capture s'effectue bien car lorsque l'écho est reçu par la patte P1.26 le Timer Control Registe T0CR prend la valeur de Timer Counter Register T0TC.