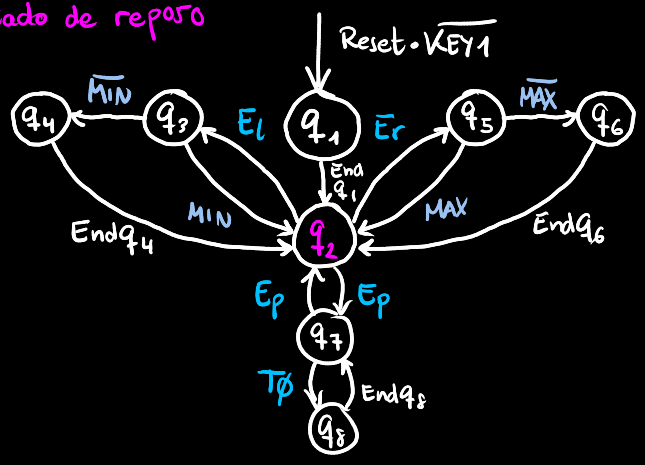
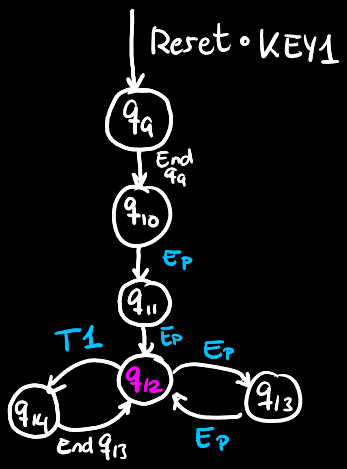
Estado de reposo






```
q1: Config_manual();
q2: while (1){
      desactivate_timer(); }
q3: LeftMovingEncoder_IRQHandler{
      check_min_reached();
q4: move_servo(-10);
q5: RightmovingEncoder_IRQHandler{
      check_max_reached(); }
```

```
q6: move_servo(10);
q7: PushEncoder_IRQHandler{
      activate_timer();
q8: Timer0_IRQ_Handler{
      IR_measure();
      display_measure();
```

```
q9: config_automatico();
q10: record_angle_left();
q11; record_angle_right();
q12; while(1){
       activate_timer();
q13: while(1){
       desactivate_timer();
     }
```

```
q14: if(direction = 0){
       move_servo(-10);}
     else{
       move_servo(10);}
     record angle();
     if(angle = MIN | angle = MAX){
       direction = direction ^ 1 // XOR}
     IR_measure();
     display_measure();
```

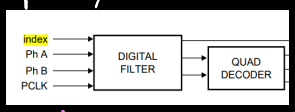
```
if KEY1 = 0{
    Config_manual(); //Pone en enable las IRS del manual
                     //Pone en disable las IRS del automatico
                     //Otras configuraciones necesarias
                     //Poner en display modo manual
}
else{
    Config_automa(); //Pone en enable las IRS del automatico
                     //Pone en disable las IRS del manual
                     //Otras configuraciones necesarias
                     //Poner en display modo manual
}
```

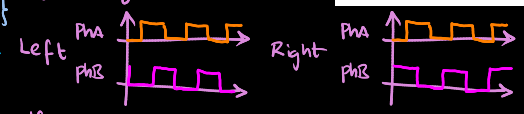
Servo → PWM0
InfraRed → ADC
Comunication → Serial
Display → Display

El: Interruption due to left encoder
MIN: Variable { if 0° → MIN, if not 0° → $\overline{MIN}$ }
Er: Interruption due to right encoder
MAX: Variable { if 180° → MAX, if not 180° → $\overline{MAX}$ }
Ep: Interruption due to push encoder*
Tø: Interruption due to Timer0
T1: Interruption due to Timer1

* Nota: Al entrar en Push_encoder_IRQHandler(); se decidirá el comportamiento según el estado desde el cual se haya mandado. Para ello se usará una variable global que definirá el estado anterior y al entrar a los estados con esa interrupción, la variable se cambiara acorde al nuevo estado.

Periferals & External interruptions:
Left moving encoder IRQ
Right moving encoder IRQ






```
If PhA =(_ or ‾) {
    if(PhB == PhA) { Pending → EINT1 // LeftMoving Encoder IRQ
    else { Pending → EINT2 // RightMoving Encoder IRQ
record_angle(){
  angle = * POSITION COUNTER REGISTER;
```

Timer0 → Timer0IRQHandler();
Timer1 → Timer1IRQHandler();