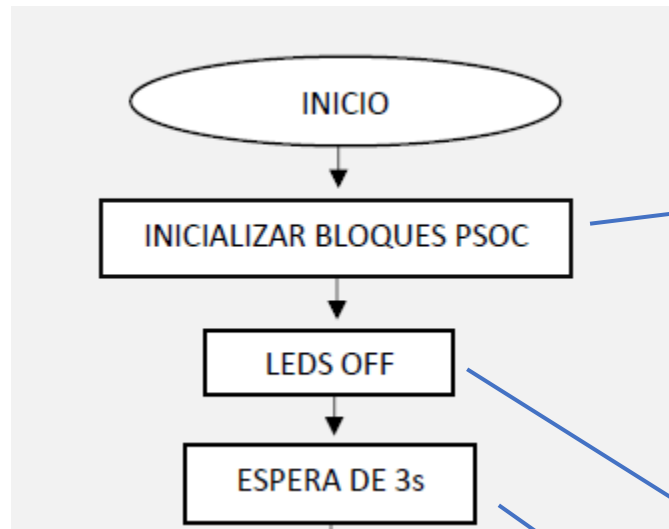


Configuración Inicial



```
CyGlobalIntEnable; // Habilita interrupciones globales
```

```
// Inicializa y configura Módulos PWM Motores A y B
```

```
PWM_1_Start();
```

```
PWM_2_Start();
```

```
// Inicializa y configura Módulo Timer y PWM Sensor Ultrasonidos
```

```
Counter_Start();
```

```
PWM_3_Start();
```

```
// Habilita Interrupción del Timer para medir el tiempo de Echo - Ultrasonidos
```

```
Counter_Int_StartEx(Counter_Interruption);
```

```
// Inicializa Módulo ADC Sensor IR
```

```
ADC_Start();
```

```
//Inicializamos leds apagados
```

```
Pin_Verde_Write(LED_OFF);
```

```
Pin_Rojo_Write(LED_OFF);
```

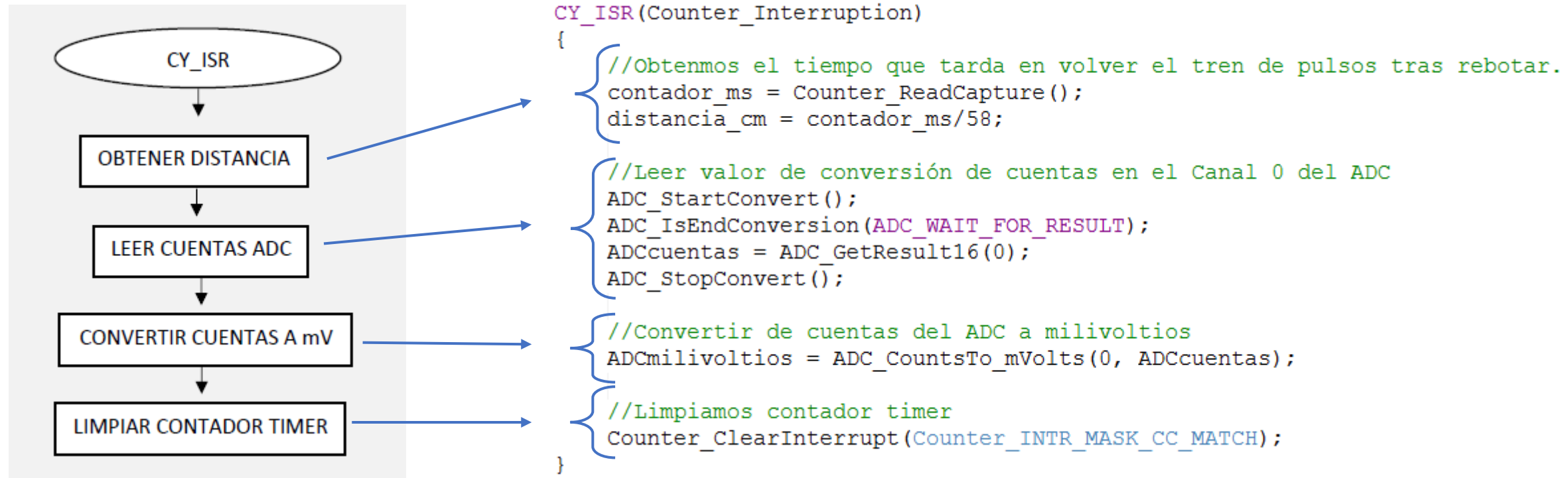
```
Pin_Azul_Write(LED_OFF);
```

```
// Delay para que el robot realice las primeras medidas de distancia y reflexión IR
```


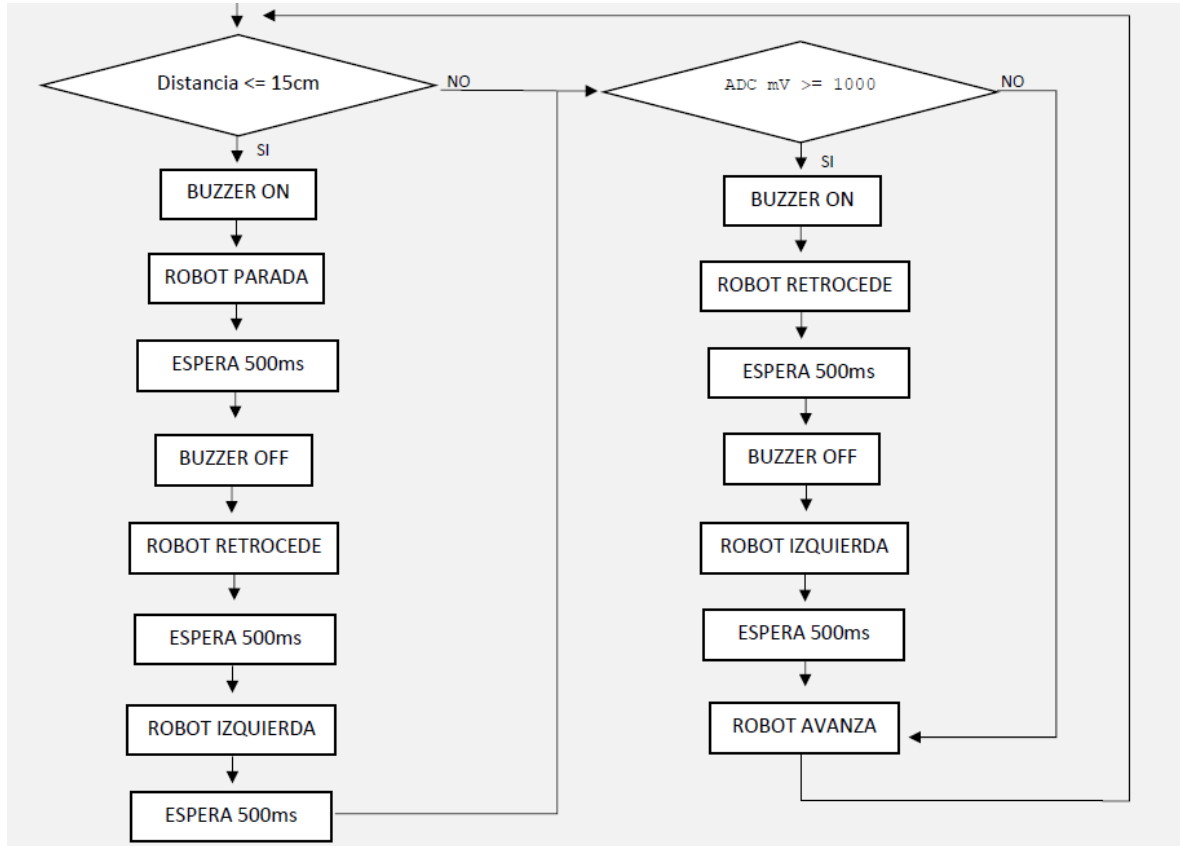
```
CyDelay(3000);
```



Interrupción



Superbucle



```
for(;;) // Superbucle
{
    if(distancia_cm <= distancia_cambio)
    {
        //Paramos 0.5 segundos y mientras pitamos.
        buzzer_pitidoEncendido();
        robot_parada();
        CyDelay(500);
        buzzer_pitidoApagado();

        //Retrocedemos 0.5 segundos.
        robot_retrocede();
        CyDelay(500);

        //Giramos 0.5 segundos.
        robot_izquierda();
        CyDelay(500);
    }
    if(ADCmilivoltios >= 1000)
    {
        //Retrocedemos 0.5 segundos y mientras pitamos.
        buzzer_pitidoEncendido();
        robot_retrocede();
        CyDelay(500);
        buzzer_pitidoApagado();

        //Giramos 0.5 segundos.
        robot_izquierda();
        CyDelay(500);
    }
    robot_avanza();
}
```

Funciones de giro

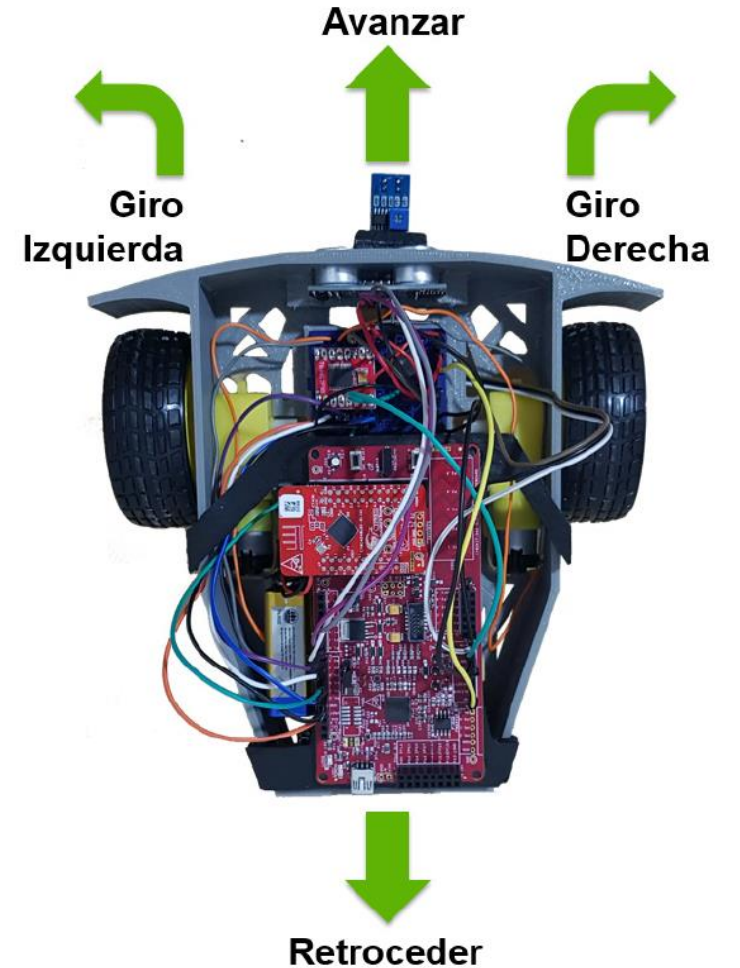


```
void robot_derecha (void) // Implementación de la función de giro derecha
{
    Pin_Verde_Write(LED_ON); //LED Amarillo
    Pin_Rojo_Write(LED_ON); //LED Amarillo
    Pin_Azul_Write(LED_OFF); //Apago el pin que no se usa para no mezclar colores

    //Configuración Girar Robot hacia la Derecha
    Pin_MotorA_A_Write(0); //Motor A retrocede
    Pin_MotorA_R_Write(1); //Motor A retrocede
    Pin_MotorB_A_Write(1); //Motor B avanza
    Pin_MotorB_R_Write(0); //Motor B avanza

    PWM_1_WriteCompare(40); //Ajusto velocidad de giro motor A al 40 %
    PWM_2_WriteCompare(40); //Ajusto velocidad de giro motor B al 40 %
}
```

Tabla Verdad Motores	Motor A			Motor B		
	P0.2	P0.1	P0.0	P0.5	P0.4	P0.3
AVANZAR	1	0	35%	1	0	35%
RETROCEDER	0	1	35%	0	1	35%
GIRO DERECHA	0	1	35%	1	0	35%
GIRO IZQUIERDA	1	0	35%	0	1	35%



Funcione zumbador



Módulo	Nombre Pin	Pin
Zumbador	Activación	P1.0



```
void buzzer_pitidoEncendido (void)
{
    Pin_Zumbador_Write(1u);
}

void buzzer_pitidoApagado (void)
{
    Pin_Zumbador_Write(0u);
}
```

