

Sensor de estacionamiento

Protocolos de Comunicación de Sistemas Embebidos

Trabajo práctico final

Autor:

- Sr. Juan Manuel Hernández

Profesor:

- Ing. Israel Pavelek

Link al repositorio: https://github.com/juanmaher/PdM_cese_workspace/tree/main/parking_sensor

Índice

Recorrido por la presentación

Guía

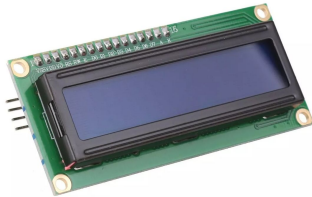
1. Aplicación
2. Periféricos
3. Máquina de estados finitos
4. Módulos
5. Detalles de código



Aplicación

Sensor de estacionamiento

- Desarrollo sobre la plataforma NUCLEO-F429ZI de ST.
- Lectura de proximidad a objetos sólidos mediante el sensor ultrasónico (HC-SR04) y el cálculo temporal del ancho de los pulsos.
- Procesamiento de los datos medidos para darle un valor de acuerdo a la resolución preestablecida (8 niveles).
- Detección de la señal de marcha atrás o reversa, representada mediante un botón.
- Escritura a una terminal serie de los valores medidos a través del protocolo UART para debugging.
- Escritura mediante I2C de los valores obtenidos en el display (PCF8574 - HD44780).



Periféricos

Periféricos

UART

- USART3
- 115200 - 8N1
- TX -> PD8
- RX -> PD9

I2C

- I2C1
- SCL -> PB8
- SDA -> PB9
- 100 kHz

GPIO

- TRIGGER PIN -> PE2
- REVERSE PIN -> PC13

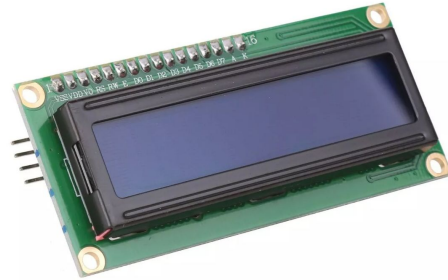
TIM

- TIM1
- ECHO PIN -> PE9
- TIM1_CC_IRQn (ambos flancos)

Display LED - Expander de I2C a 8 bit

Características del display

- Modelo: HD44780
- Modo: 4 bits - 2 lines
- Comunicación paralela

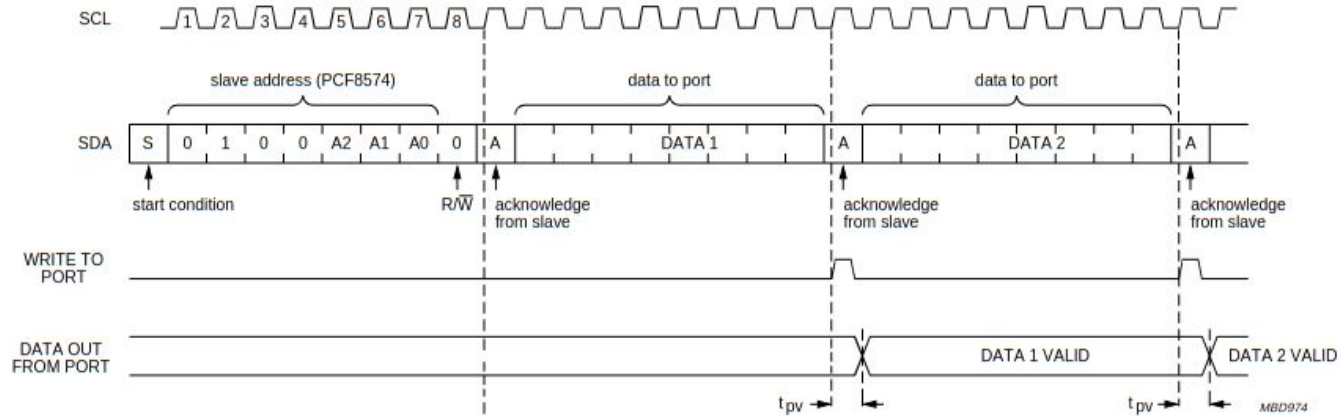


Características del expander

- Modelo: PCF8574
- Comunicación por I2C
- Frecuencia de clock: 100 kHz



Interfaz I2C



En este ejemplo, se muestra como inicialmente se envía la dirección del esclavo y luego los bits de datos que se quieren escribir. Luego del ACK, se ve como se transmiten esos datos a los pines de salida del expander.

Comunicación con Display HD44780

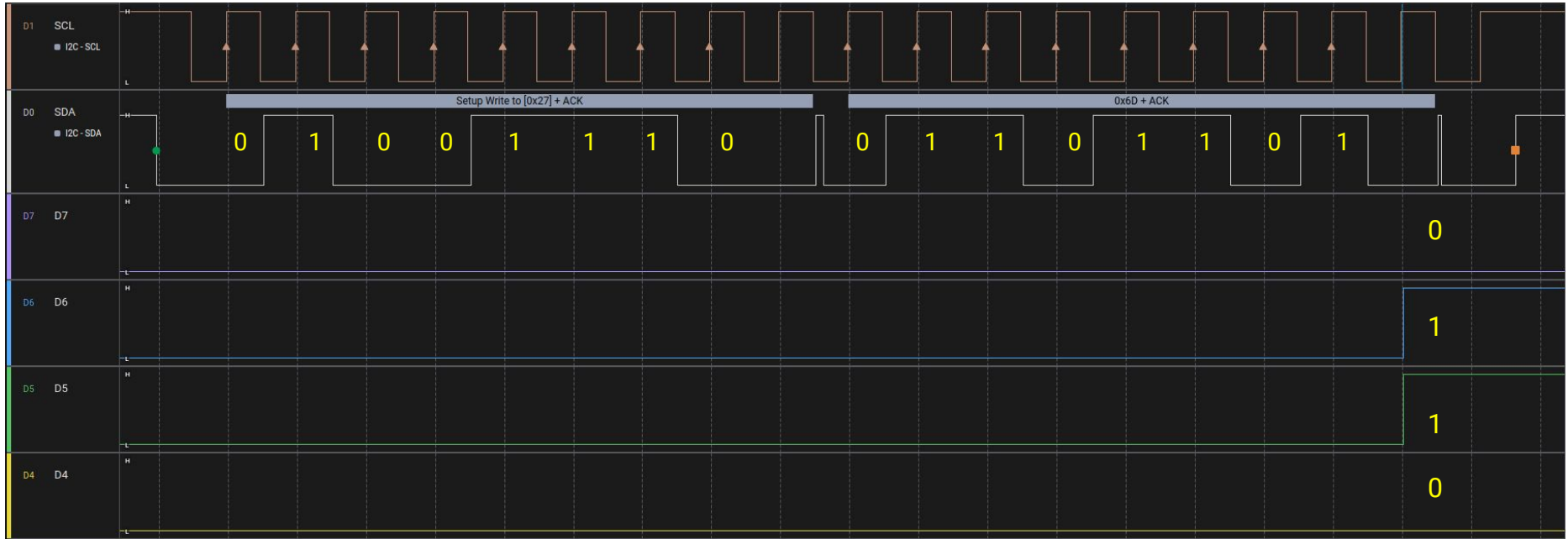
Busco que el expander escriba en su salida los 8 bits con el formato de la tabla.

Ejemplo:

0x6D = 0b01101101
D7 = 0
D6 = 1
D5 = 1
D4 = 0
E = 1
B = 1
R/~W = 0
RS = 1

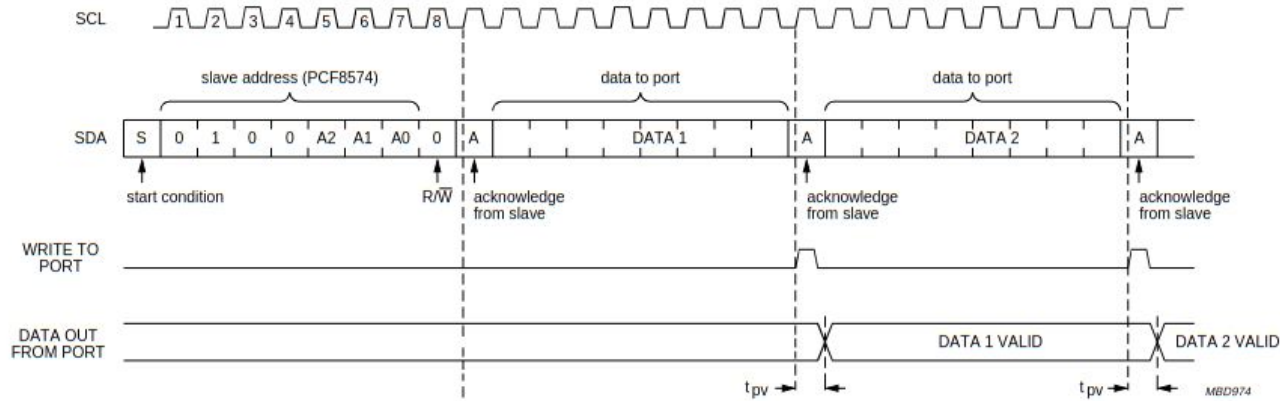
Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/ \overline{W}	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.

Análisis de bits



Primero, envió la dirección del esclavo ($0x27 \ll 1 = 0x4E = 0b01001110$). Luego, envió un comando de escritura.

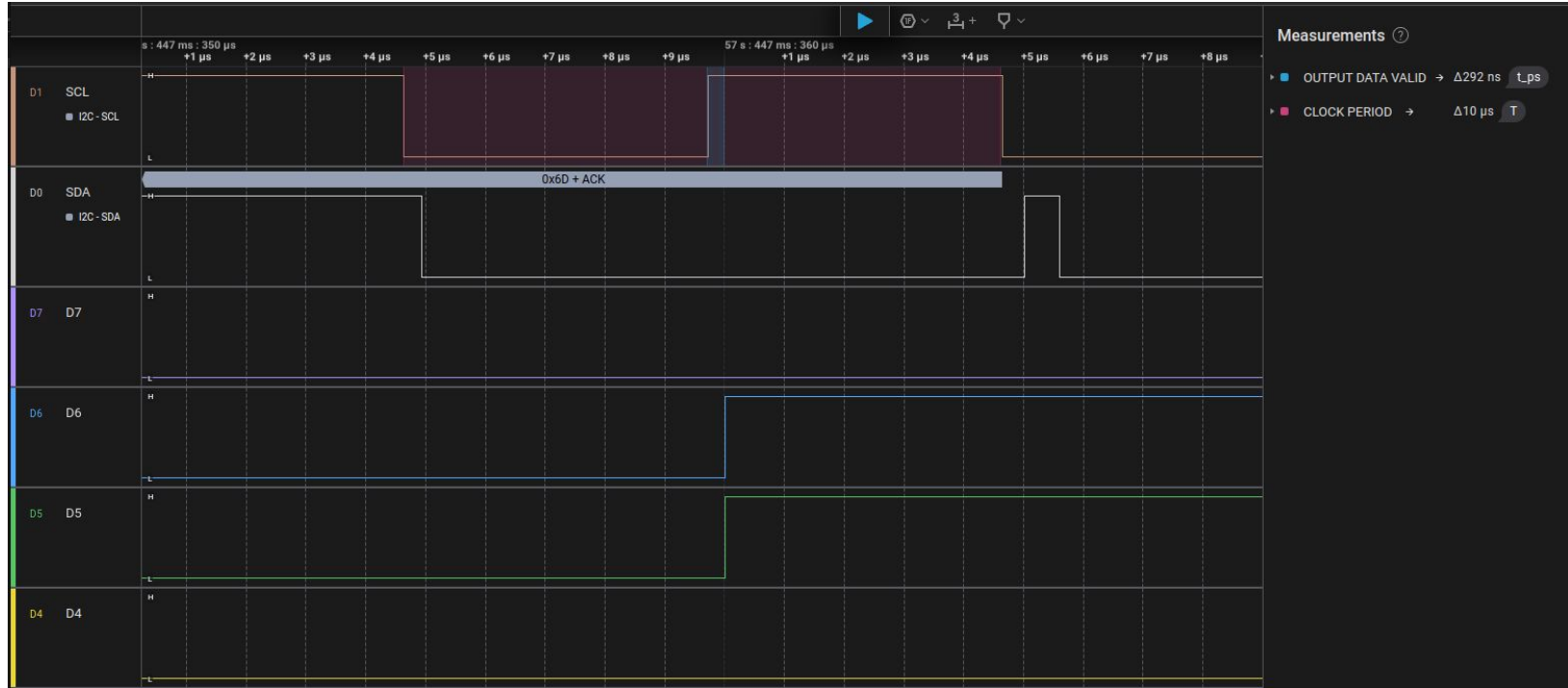
Análisis temporal de la comunicación



Symbol	Parameter	Conditions	MIN.	TYP.	MAX	UNIT
t _{pv}	output data valid	CL ≤ 100 pF	-	-	4	us

Tiempo de escritura de datos sobre pines de salida.

Medición temporal de la comunicación expander - display

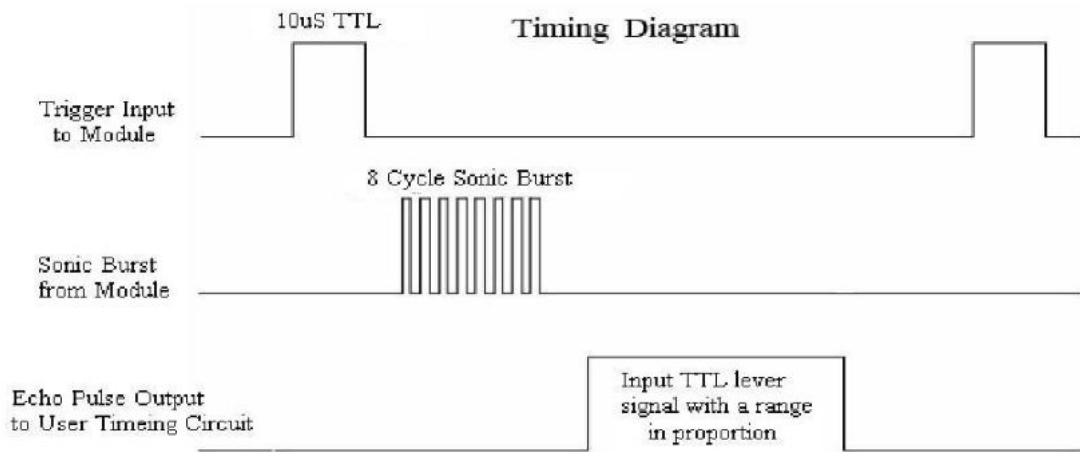


Valor máximo = 4 us y Valor medido = 0,292 us

Sensor ultrasónico

Características del sensor

- Modelo: HC-SR04
- Rango: 2 cm - 400 cm
- Precisión: +- 0.3 cm



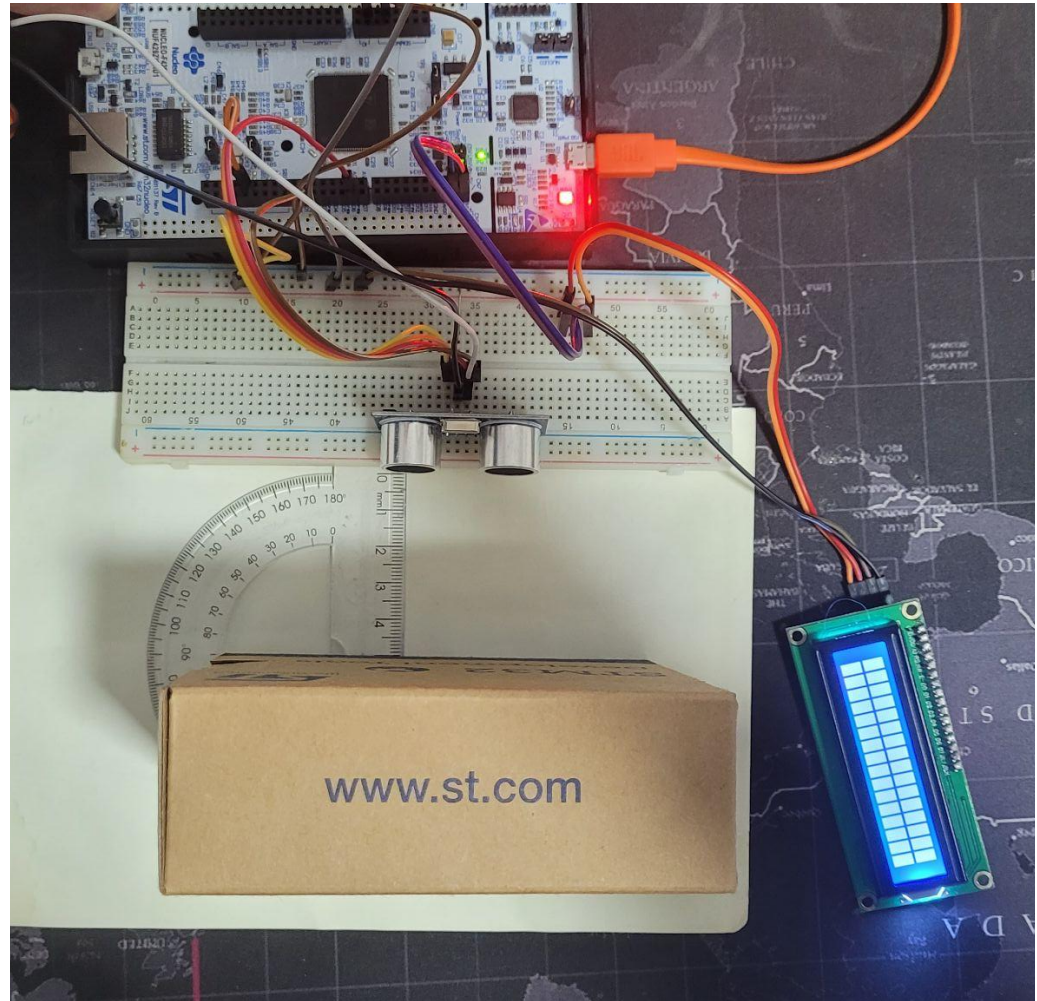
Implementación

- Módulos usados:
 - GPIO para TRIG
 - TIM para ECHO

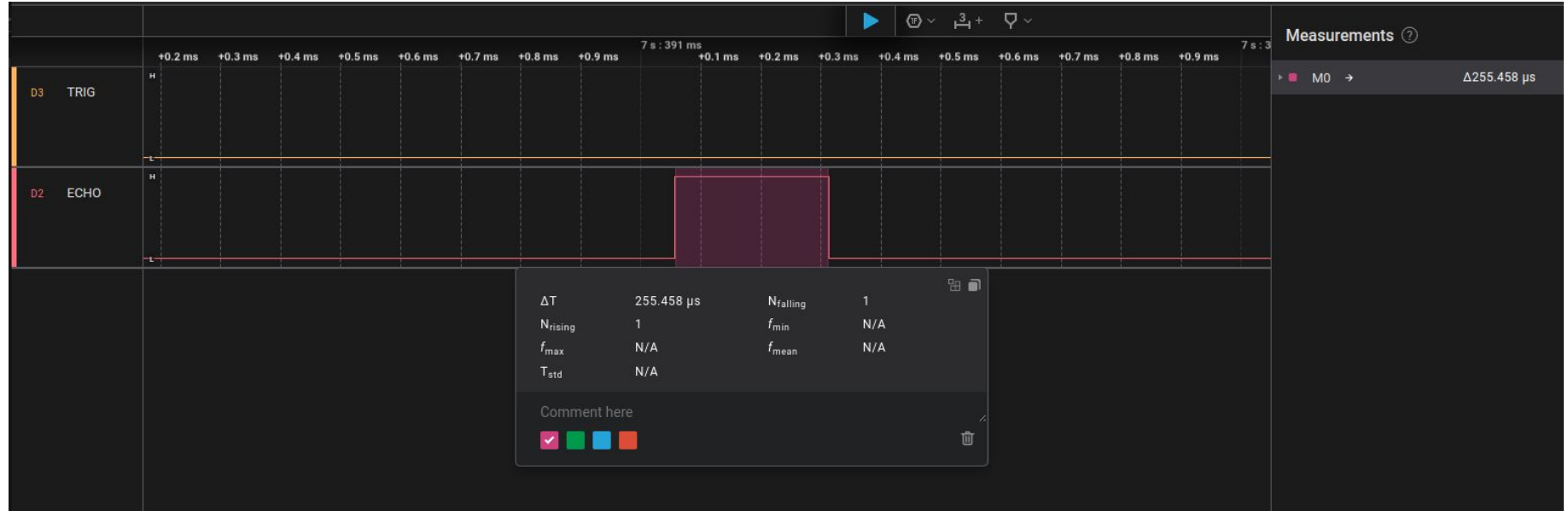


Prueba de medición

Esta medición fue tomada a aproximadamente 5 cm de un objeto. El valor obtenido fue procesado y se obtuvo el nivel máximo de resolución posible (NIVEL 8). Este dato se transforma en dos barras de 16 caracteres como se muestra en el display.



Medición temporal del pulso



Medición realizada a ~ 5 cm del sensor ultrasónico


Resultados de mediciones

Fórmula para obtención de resultado:

$D = T \times V / 2$ (D = distancia[m] , T = tiempo [s], V = velocidad [m/s])

Nro. Medición	Distancia objeto [cm]	Tiempo de pulso [us]	Distancia medida [cm]	Error [cm]
1	3	141,708	2,409	-0,591
2	5	255,458	4,343	-0,657
3	8	501,833	8,531	+0,531

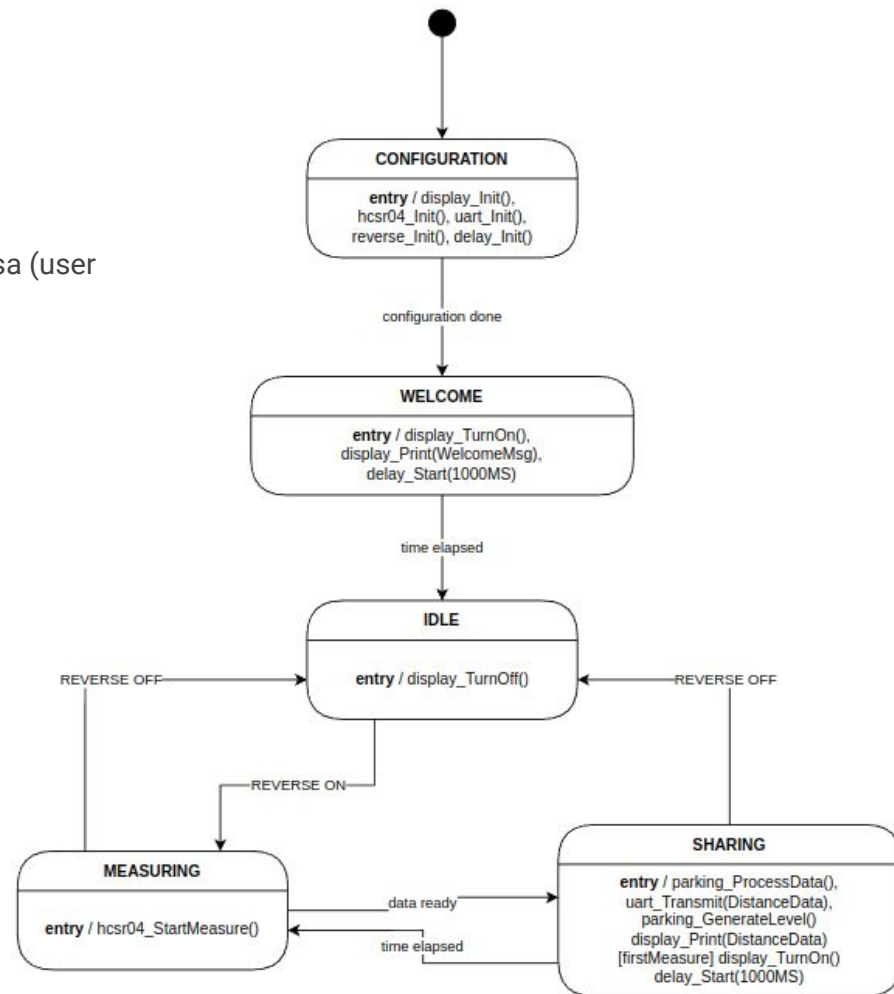
Para la ecuación se utilizó el valor de la velocidad del sonido (340 m/s). La precisión especificada por fabricante es de 0,3 cm.



Máquina de estados finitos

Máquina de estados finitos

- **CONFIGURATION**
 - Inicializo display, sensor ultrasónico, UART, señal de reversa (user button) y timers.
- **WELCOME**
 - Prendo display.
 - Escribo mensaje de bienvenida.
 - Inicio 1s timer.
- **IDLE**
 - Apago display.
- **MEASURING**
 - Inicio la medición del sensor ultrasónico.
- **SHARING**
 - Transformo la información recibida a nivel de resolución.
 - Envío el dato por UART y lo escribo en el display.
 - Prendo el display si es la primera medición.
 - Inicio 1s timer.



Módulos

Módulos

- Se separó en módulos de acuerdo a funcionalidades comunes.
 - Comunicación con el exterior (UART).
 - Interfaz de usuario (Display).
 - Herramientas y utilidades (debounce y delay no bloqueante).
 - Sensado de señales externas (ultrasónico y reversa).
- Los drivers del display, ultrasónico y reversa fueron planteados de forma genérica para no tener dependencias con el hardware. Se implementó una capa de aplicación y una capa de bajo nivel (port.c).

```

└─ Drivers
  └─ API
    └─ Inc
      └─ Communication
        └─ API_uart.h
      └─ HMI
        └─ API_display_port.h
        └─ API_display.h
      └─ Misc
        └─ API_debounce.h
        └─ API_delay.h
      └─ Sensors
        └─ API_hcsr04_port.h
        └─ API_hcsr04.h
        └─ API_reverse_port.h
        └─ API_reverse.h
    └─ Src
      └─ Communication
        └─ API_uart.c
      └─ HMI
        └─ API_display_port.c
        └─ API_display.c
      └─ Misc
        └─ API_debounce.c
        └─ API_delay.c
      └─ Sensors
        └─ API_hcsr04_port.c
        └─ API_hcsr04.c
        └─ API_reverse_port.c
        └─ API_reverse.c
```

¡Gracias!