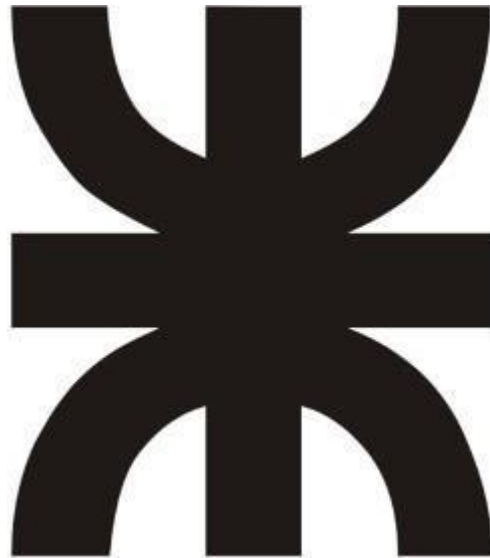


Universidad Tecnológica Nacional

Facultad Regional Tucumán



Técnicas Digitales II 2023

Sistema de Medición de Nivel de Agua
con PSoC6

Integrante:

- BOUSO PATRICIO JEREMIAS

Introducción

Este informe detalla el desarrollo de un proyecto que utiliza la placa PSoC6 de Infineon para implementar un sistema simulador de medición de nivel de agua. La placa PSoC6 se integra con un display 16x2 1602A mediante un módulo I2C, un LED, y dos botones, creando una interfaz interactiva que simula el funcionamiento de un medidor de líquido.

Objetivo

El objetivo principal de este proyecto es facilitar el uso y demostrar la capacidad de la placa PSoC6 para controlar dispositivos externos y crear una interfaz interactiva. Se simula un medidor de nivel de agua, donde la placa controla un LED que actúa como bomba de agua, un display que muestra información relevante y dos botones que interactúan con el sistema.

Componentes utilizados

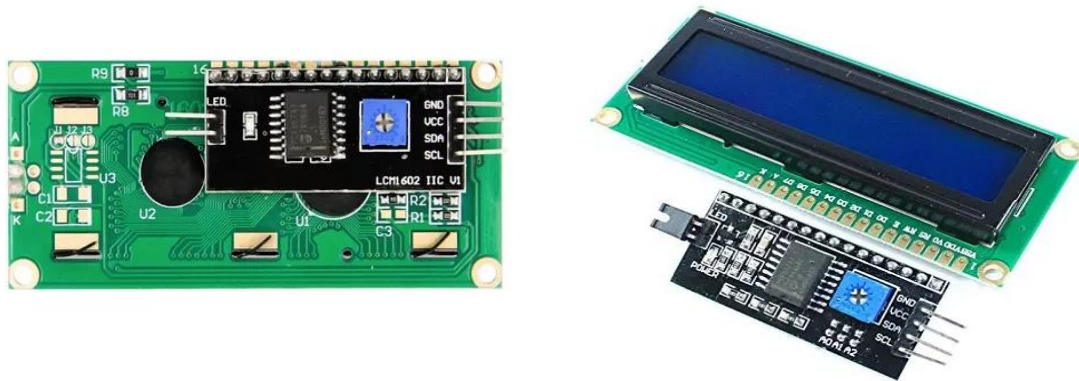
- Kit de desarrollo de prototipos PSoC™ 6 Wi-Fi BT (CY8CPROTO-062-4343W)
- Display 16x2 1602A Ver 5.5
- Modulo I2C HW-61
- 2 Pulsadores
- 1 Diodo led
- 3 Resistencias - (100 ohm, 560ohm)
- Protoboard y cables para la conexión

Software de desarrollo

- [Entorno de Desarrollo ModusToolbox de Infineon](#)

Conexión del Display

*El display y el modulo se los puede conseguir tanto por separado como ya ensamblado uno al otro



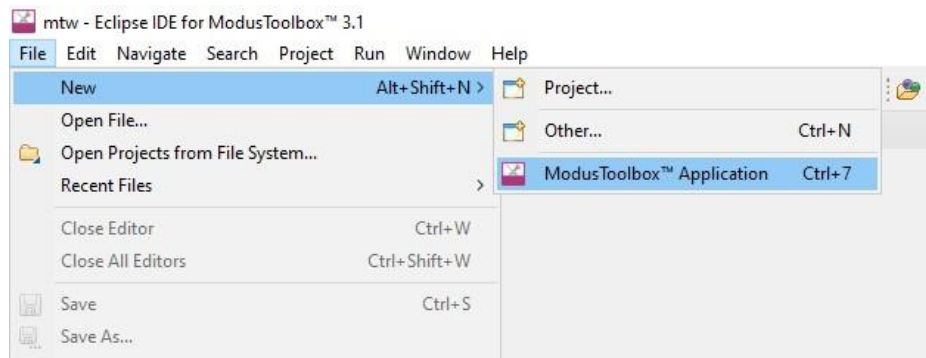
Los pines disponibles en este componente son:

- **GND**: Conexión a tierra
- **VCC**: Conexión a tensión positiva (3.3V a 5V)
- **SCL**: Es el pin de reloj para la sincronización.
- **SDA**: Es el pin de datos para la transmisión de información.

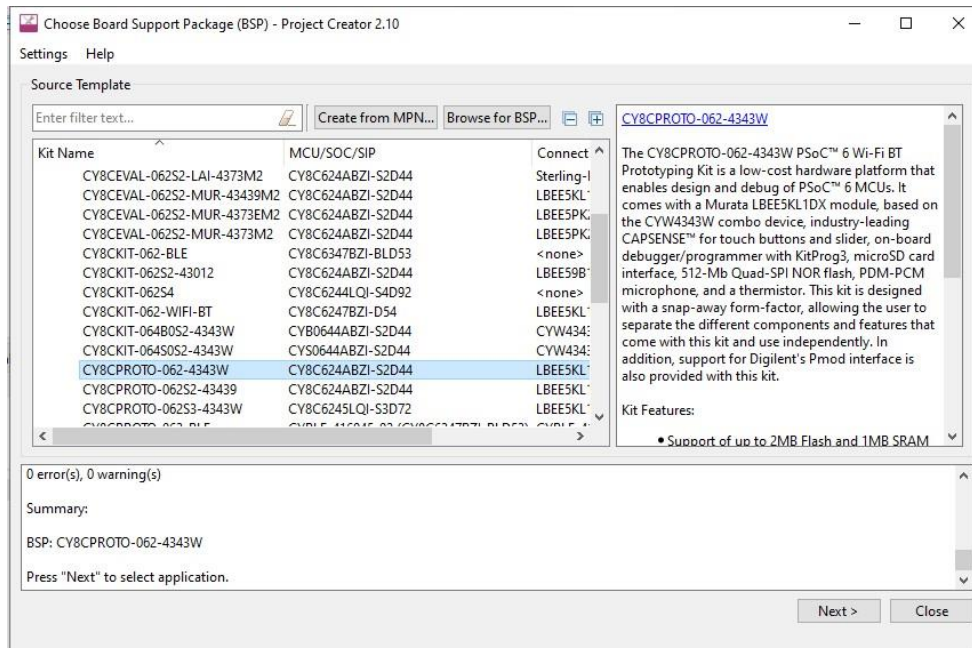
En este proyecto, se ha empleado la interfaz I2C para la conexión del display 16x2 1602A, permitiendo una comunicación eficiente entre la placa PSoC6 y el display. Además, se ha integrado un LED para representar visualmente el estado de la bomba de agua, y dos botones para la interacción del usuario con el sistema.

PASOS A SEGUIR PARA LA CREACION DEL CODIGO

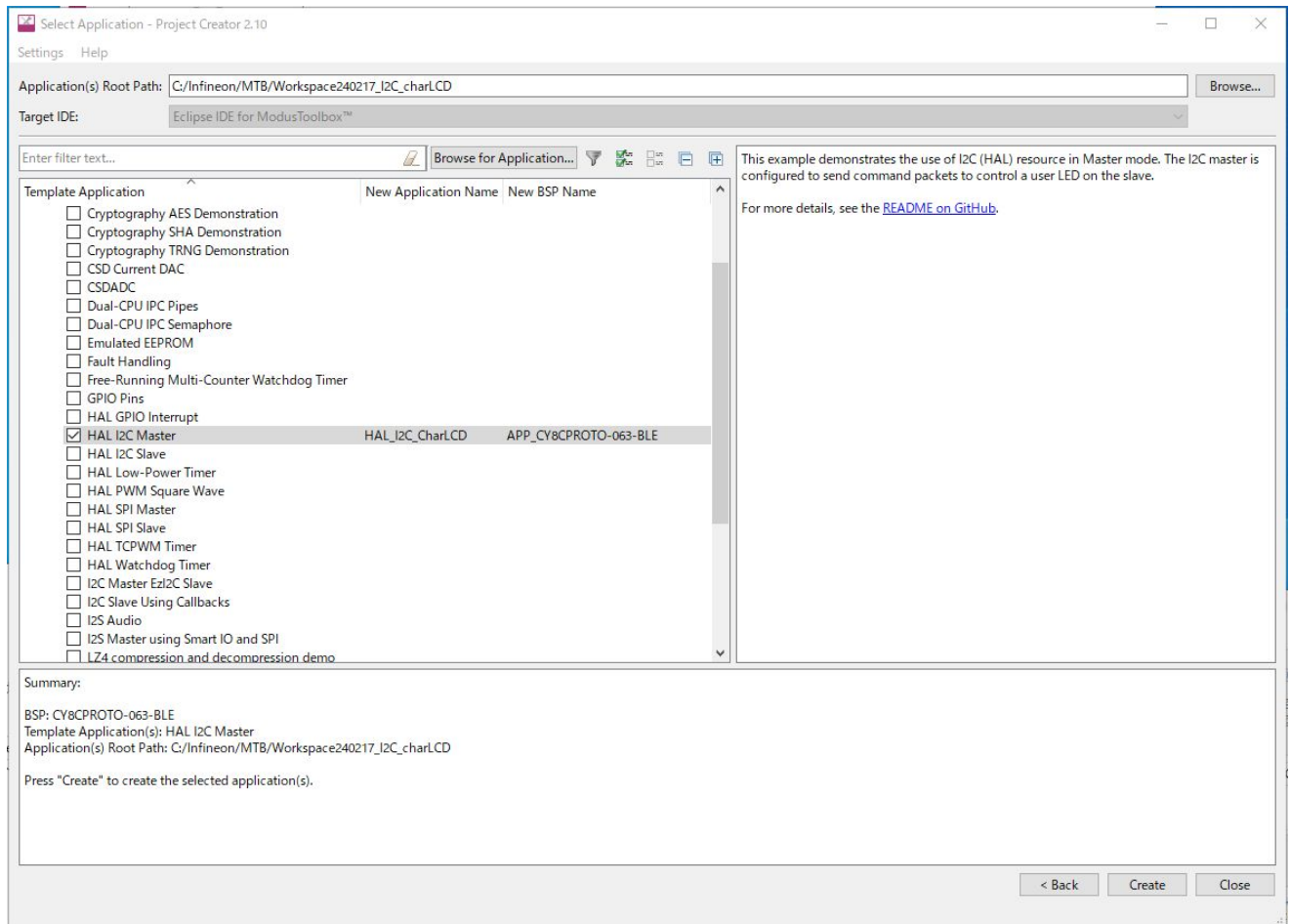
Paso 1: Creamos un nuevo proyecto en ModusToolbox, seleccionando el Paquete de Soporte adecuado para nuestra placa ([CY8CPROTO-062-4343W](#))



Paso 2: En nuestro Project Creator seleccionamos la placa que estamos utilizando, en nuestro caso es **CY8CPROTO-062-4343W**



Paso 3: Seleccionamos la opción "HAL I2C Master" de "Periféricos". Y le ponemos un nombre al proyecto. En mi caso le puse "HAL_I2C_CharLCD"



Paso 4: Una vez creada la aplicación, necesitamos importar los siguientes archivos fuente y archivos de encabezado: i2c_clcd.h, i2c_clcd.c, i2c_utils.h, i2c_utils.c, main.c

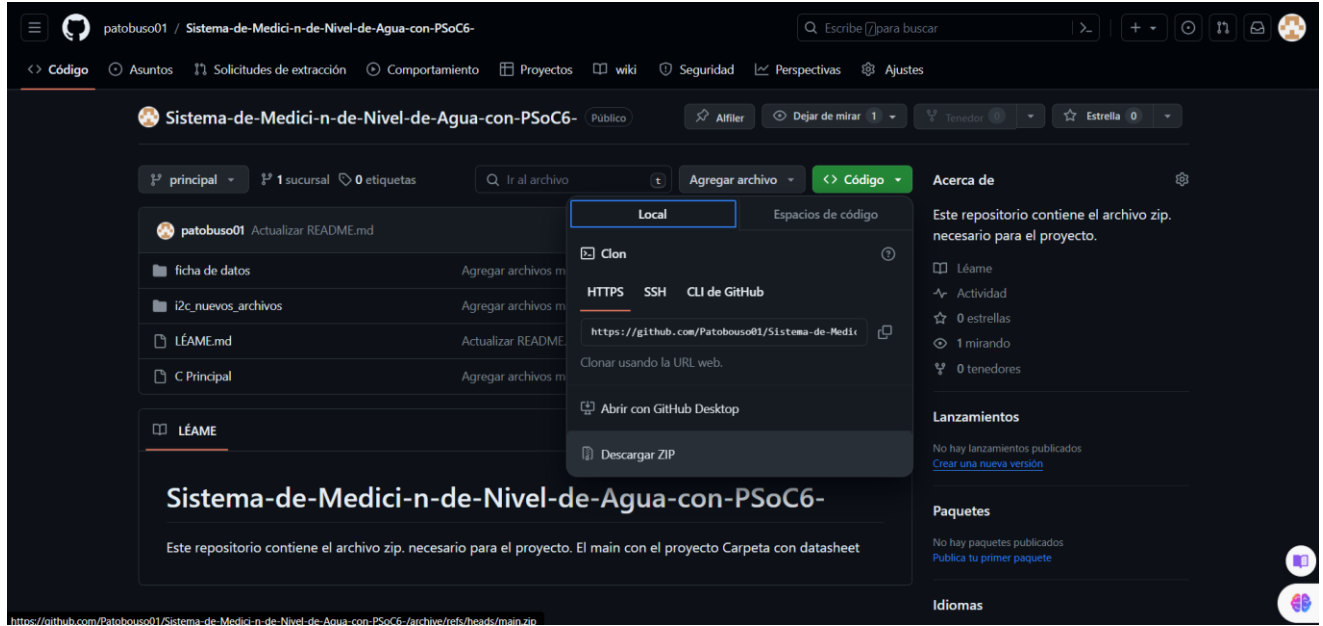
Estos archivos se encuentran en una carpeta zip. Llamada i2c_new_files en el link del repositorio que dejare al final del documento.

Simplemente copiamos y pegamos estos archivos, en la carpeta del proyecto.

El main viene con un código que mostrará "Hello" durante 2 segundos y luego comenzará a contar.

Universidad Tecnológica Nacional

Facultad Regional de Tucumán



| Nombre | Fecha de modificación | Tipo | Tamaño |
|-----------|-----------------------|-----------|--------|
| Hoy | | | |
| i2c_clcd | 28/2/2024 01:00 | Archivo C | |
| i2c_clcd | 28/2/2024 01:00 | Archivo H | |
| i2c_utils | 28/2/2024 01:00 | Archivo C | |
| i2c_utils | 28/2/2024 01:00 | Archivo H | |
| main | 28/2/2024 01:00 | Archivo C | |

Copiar (Ctrl+C)

✂️ 📄 📄 📄 🗑️

📄 Abrir

🔗 Compartir

★ Agregar a Favoritos

📦 Comprimir en archivo ZIP

📄 Copiar como ruta de acceso

🔑 Propiedades

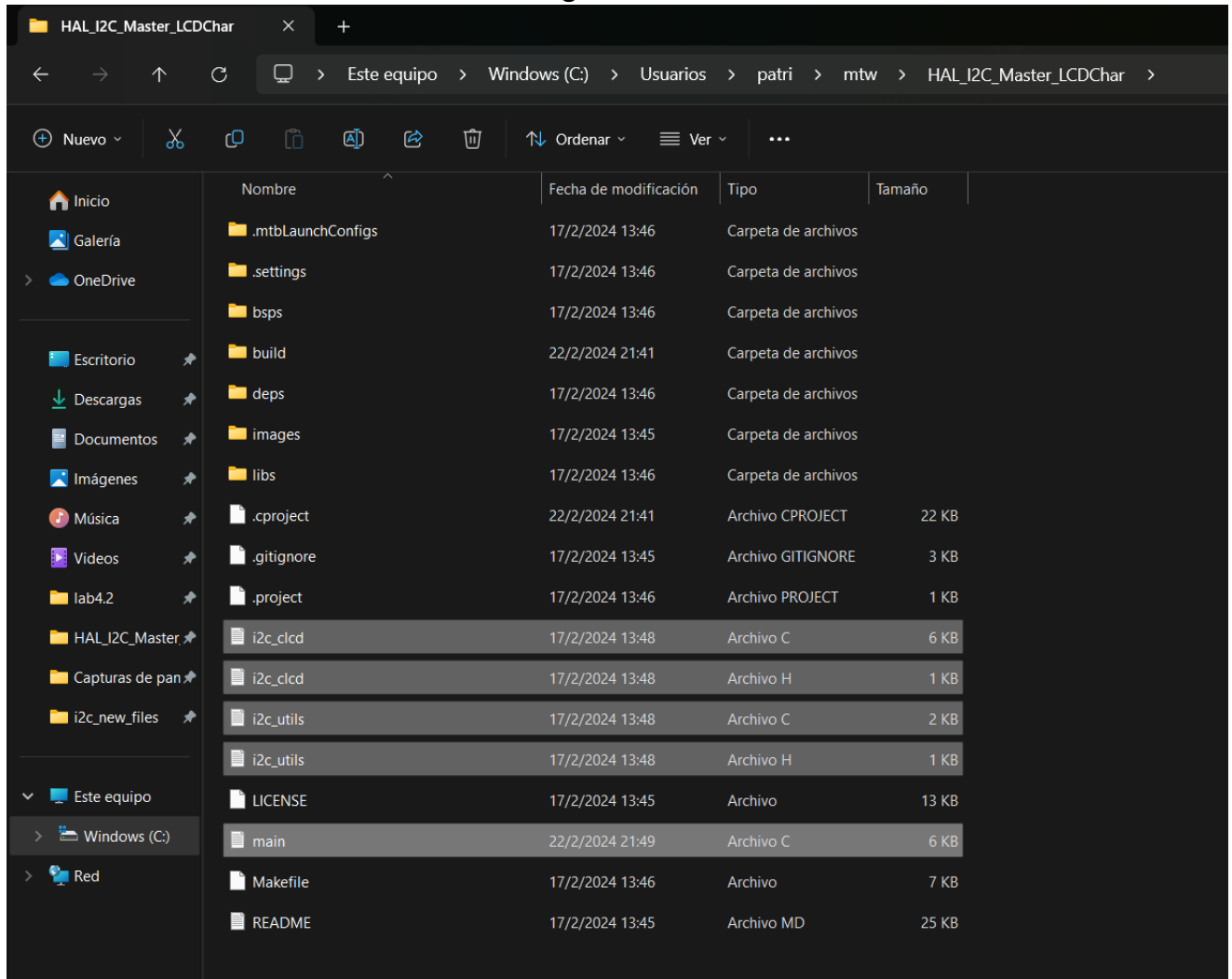
📄 Editar en el Bloc de notas

🗜️ WinRAR

🔗 Mostrar Más opciones

Universidad Tecnológica Nacional

Facultad Regional de Tucumán



Paso 5: Antes de proceder a editar nuestro código, debemos establecer los puertos y pines donde se conectarán el display, el led y los botones en nuestra PSoC6.

ESQUEMA ELECTRICO

En mi caso conecte

DISPLAY

VCC= 3.3 V

GND= GND

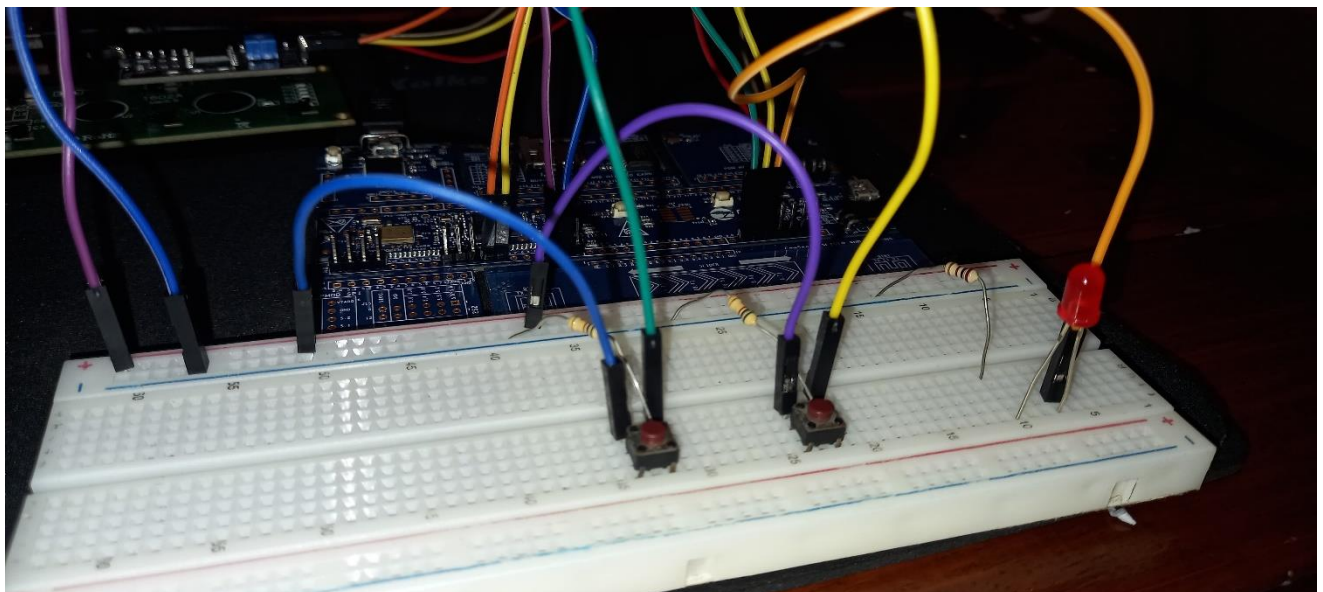
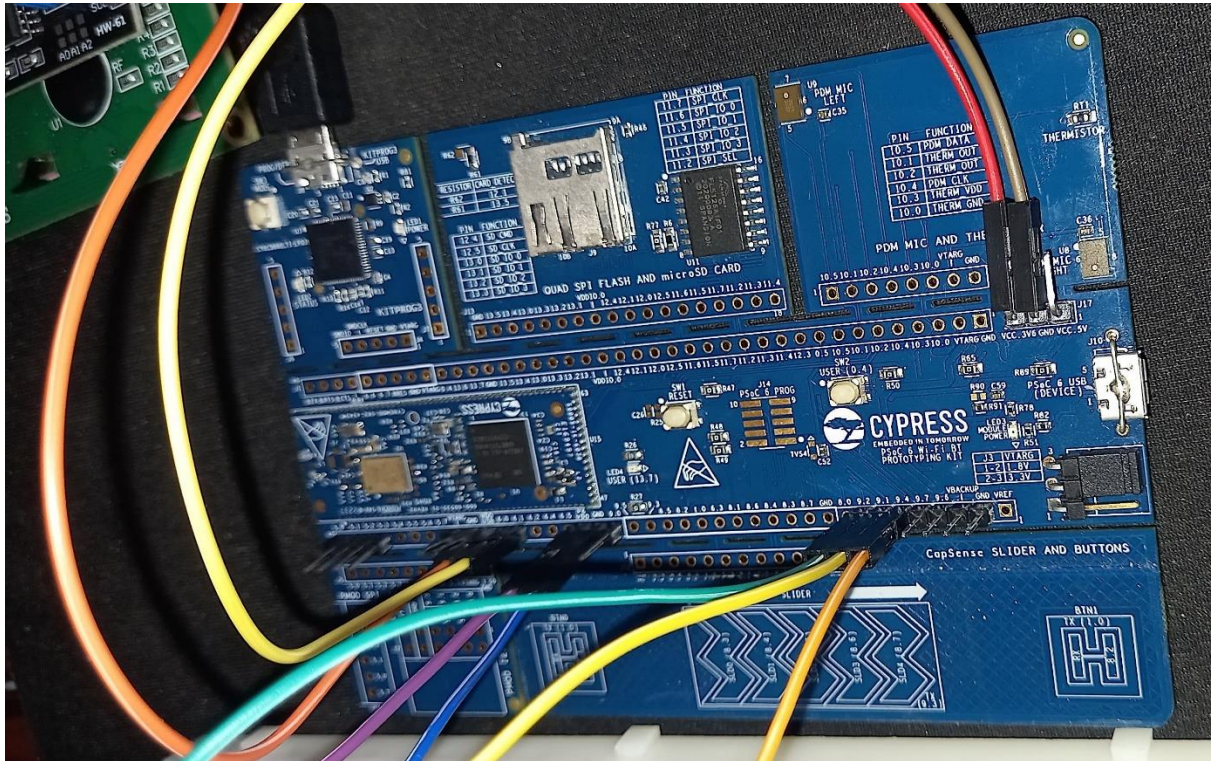
SCL = P6[0]

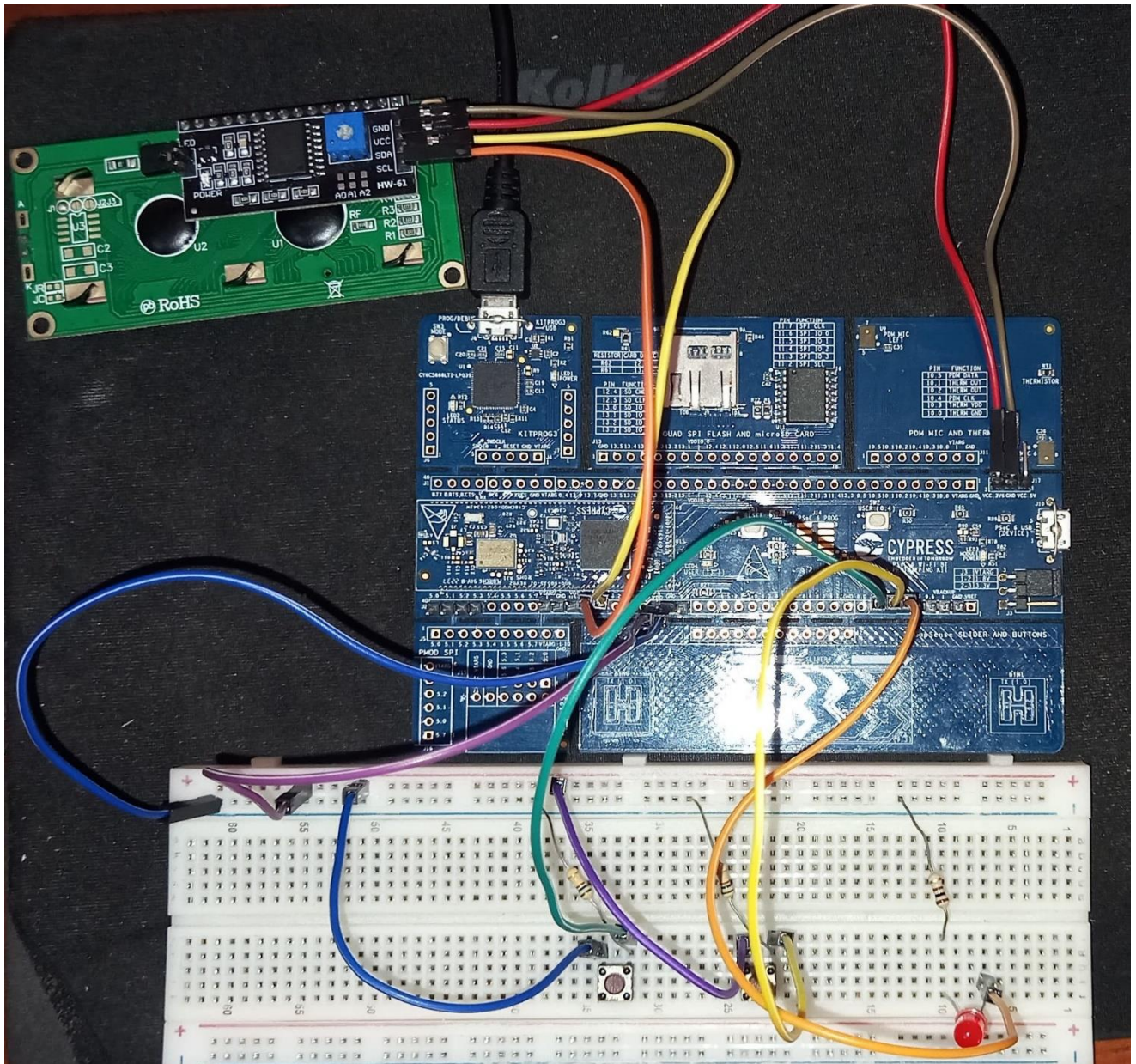
SDA = P6[1]

BOTON BAJO = P9[2]

BOTON ALTO = P9[1]

LED= P9[4]





Paso 6: Ingresamos a main, y copiamos el siguiente código donde se especifica los pines de la placa a utilizar, y también la programación para mostrar los mensajes en la pantalla y la programación de los botones.

Universidad Tecnológica Nacional
Facultad Regional de Tucumán

```
*****/
#include "cyhal.h"
#include "cybsp.h"
#include "cy_retarget_io.h"
#include "i2c_utils.h"
#include "i2c_clcd.h"

/*****
 * Macros
 *****/
/* Delay of 1000ms between commands */
#define CMD_TO_CMD_DELAY    (1000UL)

/* Command valid status */
#define STATUS_CMD_DONE     (0x00UL)

/* Packet size */
#define PACKET_SIZE         (3UL)

void handle_error(uint32_t status)
{
    if (status != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(0);
    }
}

void init_hardware(void)
{
    cy_rslt_t result;

    /* Initialize the device and board peripherals */
    result = cybsp_init();
    /* Board init failed. Stop program execution */
    handle_error(result);

    /* Initialize the retarget-io */
    result = cy_retarget_io_init(CYBSP_DEBUG_UART_TX, CYBSP_DEBUG_UART_RX,
                                CY_RETARGET_IO_BAUDRATE);
    /* Retarget-io init failed. Stop program execution */
    handle_error(result);

    // result = i2c_init() ;
    // handle_error(result) ;

    /* Enable interrupts */
    __enable_irq();
}

int main(void)
```

Universidad Tecnológica Nacional
Facultad Regional de Tucumán

```
{

init_hardware() ;

/* \x1b[2J\x1b[H - ANSI ESC sequence for clear screen */
printf("\x1b[2J\x1b[H");

printf("***** "
       "HAL: I2C Master CharLCD "
       "***** \r\n\n");

I2C_CLCD_Init() ;
I2C_CLCD_PutString("CARGANDO") ;

cyhal_gpio_init(P9_2,CYHAL_GPIO_DIR_INPUT, CYHAL_GPIO_DRIVE_PULLUP, 1); //BOTON
BAJO
cyhal_gpio_init(P9_1,CYHAL_GPIO_DIR_INPUT, CYHAL_GPIO_DRIVE_PULLUP, 1); //BOTON
ALTO
cyhal_gpio_init(P9_4,CYHAL_GPIO_DIR_OUTPUT, CYHAL_GPIO_DRIVE_STRONG, true);
//LED
bool    read_val1;
bool    read_val2;

for (;;)
{
    read_val1 = cyhal_gpio_read(P9_2);
    read_val2 = cyhal_gpio_read(P9_1);
    if(read_val1 == 1){
        I2C_CLCD_Clear();
        I2C_CLCD_PutString("NIVEL BAJO") ;
        cyhal_gpio_write(P9_4, 0);
        cyhal_system_delay_ms (200);
        cyhal_gpio_write(P9_4, 1);
        cyhal_system_delay_ms (200);
        cyhal_gpio_write(P9_4, 0);
        cyhal_system_delay_ms (200);
        cyhal_gpio_write(P9_4, 1);
        cyhal_system_delay_ms (200);
        cyhal_gpio_write(P9_4, 1);
    }

    if(read_val2 == 1){
        I2C_CLCD_Clear();
        I2C_CLCD_PutString("NIVEL ALTO");
        cyhal_gpio_write(P9_4, 0);
        cyhal_system_delay_ms (200);
        cyhal_gpio_write(P9_4, 1);
    }
}
```

```
cyhal_system_delay_ms (200);  
cyhal_gpio_write(P9_4, 0);  
cyhal_system_delay_ms (200);  
cyhal_gpio_write(P9_4, 1);  
cyhal_system_delay_ms (200);  
cyhal_gpio_write(P9_4, 0);  
  
}  
  
}  
}
```

La implementación de este código se basa en la simulación de un sistema de bombeo de agua. Al encenderse, el LED simula la activación de la bomba, y en el display se muestra el mensaje "CARGANDO". La interacción con los botones permite cambiar el estado del sistema:

Botón 1 (Sensor de Nivel Bajo): Al ser presionado, simula la detección de un nivel bajo de agua. El LED permanece encendido, y en el display se muestra "NIVEL BAJO".

Botón 2 (Sensor de Nivel Alto): Al ser presionado, simula la detección de un nivel alto de agua. El LED parpadea y se apaga, indicando que la bomba ha dejado de funcionar. En el display se muestra "NIVEL ALTO".

MEJORAS QUE SE PODRIAN REALIZAR EN UN FUTURO: Se puede explorar la posibilidad de mejorar la simulación agregando más funcionalidades, como la inclusión de sensores reales, tanto como sensores de líquido o sensores ultrasónicos, para detectar niveles de agua o la implementación de algoritmos más complejos para el control del sistema de bombeo.

Universidad Tecnológica Nacional

Facultad Regional de Tucumán

Todos los códigos necesarios para la realización de este proyecto se encuentran en mi link de GitHub el cual agregaremos aquí debajo, aquí encontraran la carpeta zip. El código main, carpeta con datasheet del display lcd y del modulo I2C, todo lo necesario para hacer funcionar este proyecto:

<https://github.com/Patobouso01/Sistema-de-Medici-n-de-Nivel-de-Agua-con-PSoC6-.git>

Video mostrando el funcionamiento de nuestro proyecto:

<https://youtube.com/shorts/BcgmlCbgUXQ>