

RSENSE20_ANGOSTO_TRABAJO, identificación estados lavadora.

David Angosto Latorre, 633706

Contenido

Introducción	2
Diseño.....	2
Montaje.....	6
Análisis de los datos.	8
Conclusiones	10

Introducción

La medida de las vibraciones de un equipo permite conocer no solo su estado de funcionamiento sino también su estado de salud. Para ello se suelen usar técnicas frecuenciales de análisis, aunque también es posible usar técnicas más avanzadas. Su aplicación es típicamente en ambiente industriales en los que se monitorizan grandes máquinas facilitando su mantenimiento preventivo.

Pero en el presente trabaja el campo de aplicación es de usuario final, en concreto de estudian las vibraciones de una lavadora durante un proceso de lavado de media hora. Este proceso tiene diferentes fases de lavado que implican el paro y arranque del tambor, y hay una fase final de centrifugado en el que el tambor arranca a máxima velocidad.

Para la medida de la vibración se usa un LSM6DSL, es una IMU de ST, con giróscopo y acelerómetro. Tiene una frecuencia de muestreo de hasta 6.66 KHz, e implementa filtros analógicos y digitales que aseguran no se produzca aliasing durante la medida.

Diseño

El módulo usado para la lectura del LSM6DSL es un ESP32. Este se configura para que funcione a 240 MHz y así tratar de conseguir la mayor frecuencia de muestreo posible. Para maximizar así los recursos del módulo el programa a ejecutar es lo más sencillo posible.

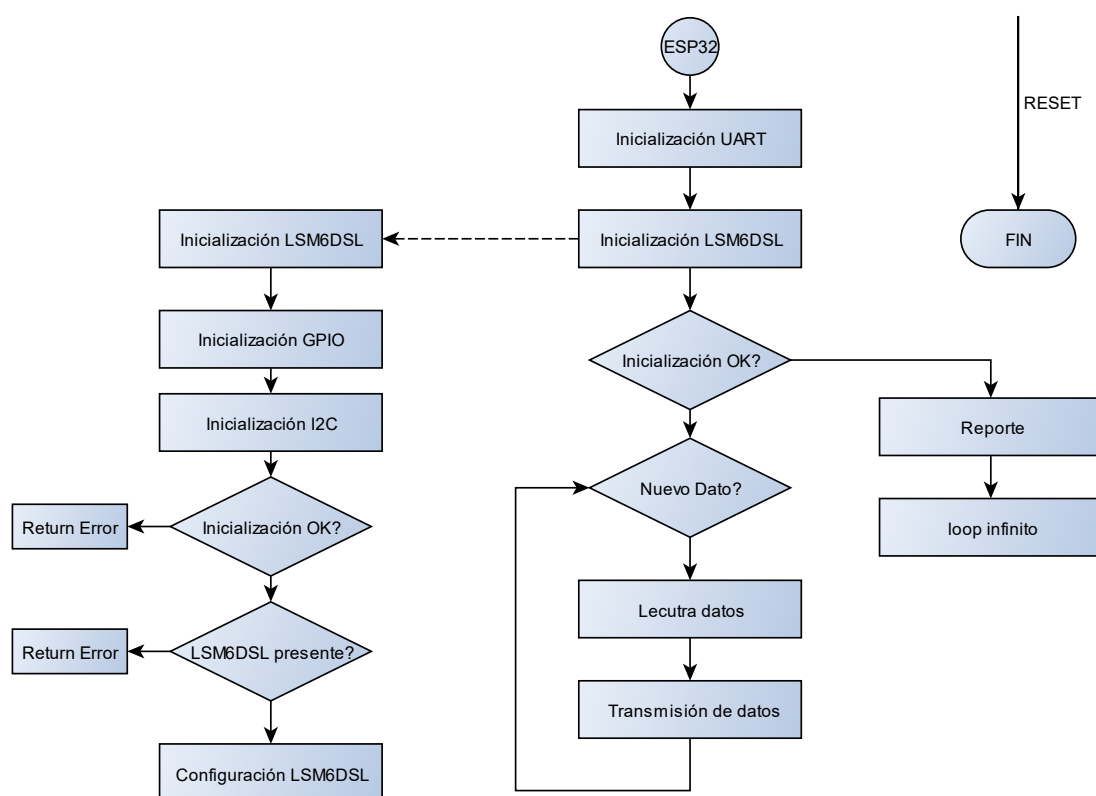


Ilustración 1: diagrama de flujo del ESP32

El método de detección de si hay un nuevo dato de aceleración disponible es con la lectura de un pin de interrupción que dispone el LSM6DSL, este se configura para que sea una 1 digital en caso de que hay un dato y es 0 cuando se ha leído. El registro además se configura para que se

machaque el data anterior en caso de que no se lea y así se asegura que se lee siempre el último dato medido.

Por su parte en un ordenador se con un script de Python se guardan los datos en un archivo .txt con tabuladores como separadores de datos. Este mensaje formatea en el propio microcontrolador por lo que el Python solo tiene que guardar el mensaje recibido sin tener que aplicar ningún formato.

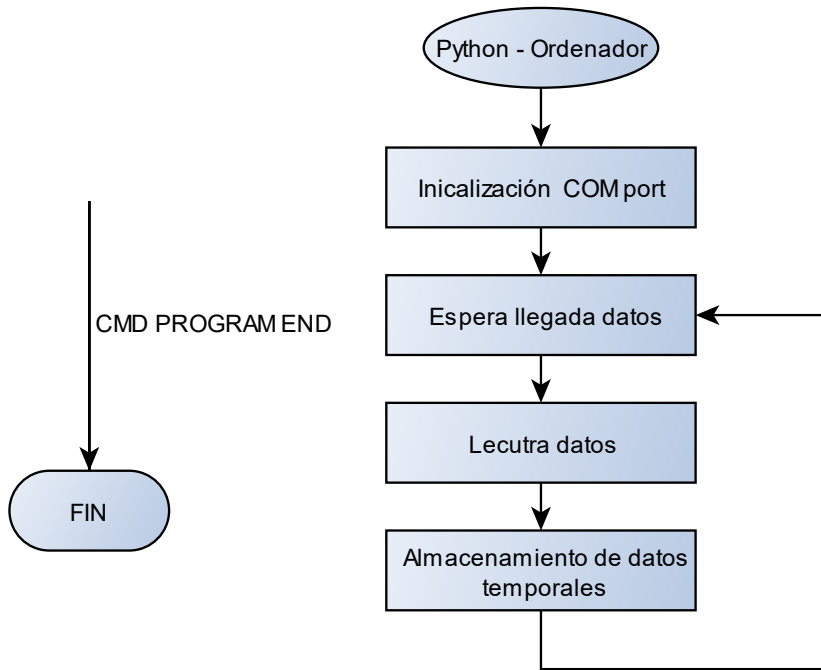


Ilustración 2: Script de Python para el almacenamiento de los datos.

El Software del ESP32 se diseña de tal manera que la aplicación software no se preocupe de ninguna configuración de bajo nivel del ESP32 ni del acelerómetro. Para ello se usa la siguiente arquitectura.

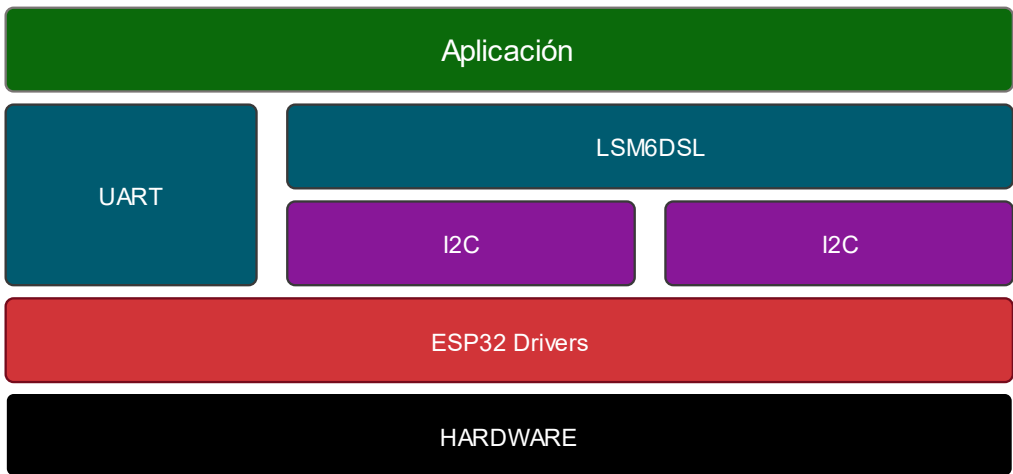


Ilustración 3: diagrama de bloques de la aquitectura del SW del ESP32.

En el ESP32 se intenta implementar la lectura en una sola transmisión de I2C a 400KHz, pero los datos recibidos no son correctos, así que se tiene que implementar en tres transmisiones esto provoca que se tarde más en la lectura de los datos.

```
esp_err_t I2C_init(void){
    i2c_config_t conf = {
        .mode = I2C_MODE_MASTER,
        .sda_io_num = SDA_PIN,
        .sda_pullup_en = GPIO_PULLUP_ENABLE,
        .scl_io_num = SCL_PIN,
        .scl_pullup_en = GPIO_PULLUP_ENABLE,
        .master.clk_speed = 400000,
    };
    esp_err_t err = i2c_param_config(I2C_NUM_0, &conf);
    i2c_driver_install(I2C_NUM_0, conf.mode, 0, 0, 0);
    return err;
}
```

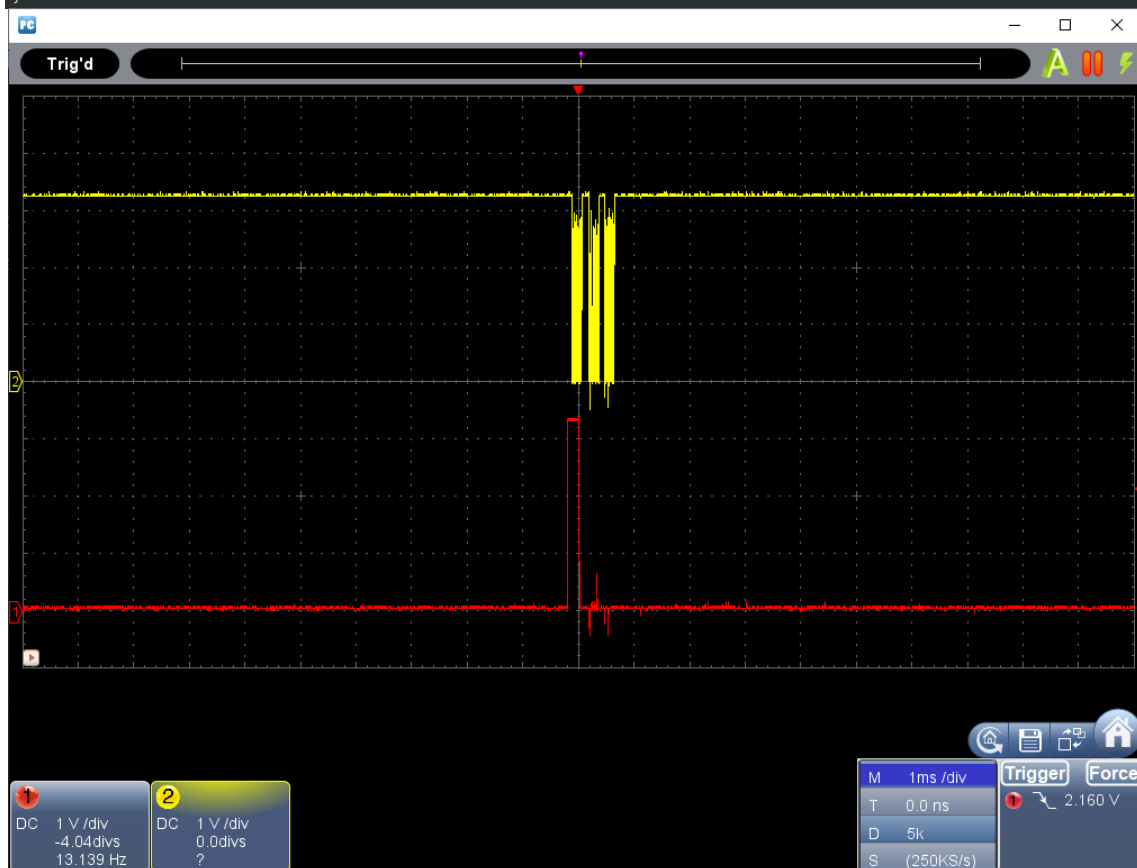


Ilustración 4Rojo: Señal de interrupción de dato listo, Amarillo, señal de I2C.

Además, el puerto de comunicación se configura con un baudio de solo 230400, por lo que sigue aumentado el tiempo de muestreo.

```
void UART_init (){
    uart_config_t PC_COM = {
        .baud_rate = 230400,
        .data_bits = UART_DATA_8_BITS,
        .parity = UART_PARITY_DISABLE,
        .stop_bits = UART_STOP_BITS_1,
        .flow_ctrl = UART_HW_FLOWCTRL_DISABLE,
        .use_ref_tick = 1,
    };
}
```

```

uart_param_config(UART_NUM_0, &PC_COM);
uart_driver_install(UART_NUM_0, 255, 255, 0, NULL, 0);

uart_set_pin(UART_NUM_0, UART_PIN_NO_CHANGE, UART_PIN_NO_CHANGE,
UART_PIN_NO_CHANGE, UART_PIN_NO_CHANGE);
}

```

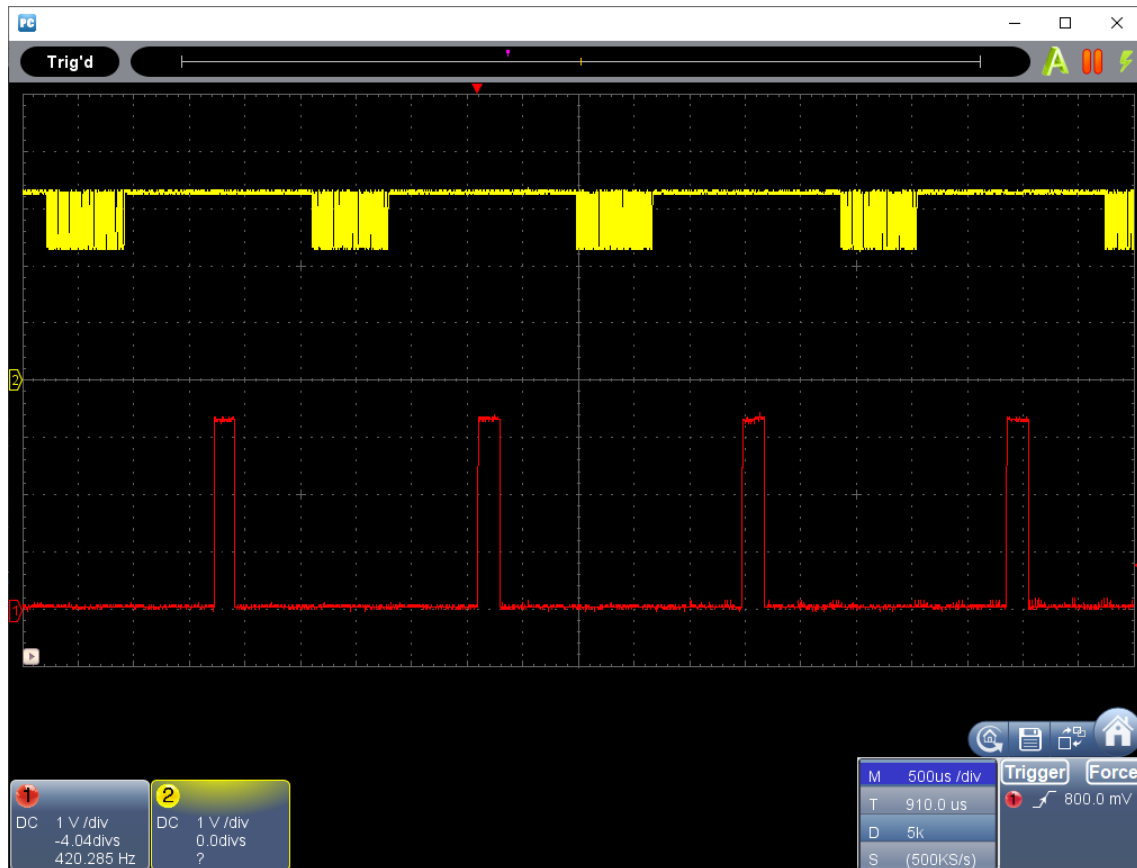


Ilustración 5: Rojo: Señal de interrupción de dato listo, Amarillo, señal de puerto serie.

Con todos estos retrasos la frecuencia de muestreo que se consigue configurar es de 416Hz así se tiene un ancho de banda de 207Hz. Aunque no se ha podido desmontar la lavadora para comprobar los datos técnicos de los motores estos suelen ser de 50Hz, así seremos capaces de ver hasta el cuarto armónico.

El LSM6DSI se configura con un filtro analógico de 1.5 KHz y uno digital de 207Hz, con una escala de $\pm 2g$, así un valor de 1g sería de 16384. No se cambia a valores de G para minimizar el tiempo de ejecución en el ESP32. Además, la función estándar `sprintf` en el ESP32 no soporta el formato de datos tipo float.

```

void LSM6DSL_configPeriodicReadings (void) {
    uint8_t data;
    data = 0x01; //Acc in FIFO no decimation
    I2C_writeRegisters(I2C_ADDRESS_HIGH, FIFO_CTRL3, &data, 1);

    data = 0x80; //Enable FIFO threshold level use
    I2C_writeRegisters(I2C_ADDRESS_HIGH, FIFO_CTRL4, &data, 1);

    data = 0b01010001; //Enable FIFO ODR 416 Hz and Continuous
    I2C_writeRegisters(I2C_ADDRESS_HIGH, FIFO_CTRL5, &data, 1);
}

```

```

//data = 0x08 //interrupt FIFO threshold
data = 0x01; //interrupt Acc data ready
I2C_writeRegisters(I2C_ADDRESS_HIGH,INT1_CTRL,&data,1);

data = 0b01100010;
/* ODR Acc & PowerMode = 416 Hz and Low power only
 * Full scale = 2g
 * Acc digital chain = enable
 * Acc analog chain bandwidth = 1.5KHz
 */
I2C_writeRegisters(I2C_ADDRESS_HIGH,CTRL1_XL,&data,1);
//by default CTRL8_XL set the low path filter with ODR/2
}

```

El diseño Hardware por su parte es mucho más sencillo, ya que solo consta de tres elementos, el ESP32, LSM6DSL y el PC que recoge los datos. EL PC alimenta al ESP32 que con su regulador de 3.3V alimenta al LSM6DSL.

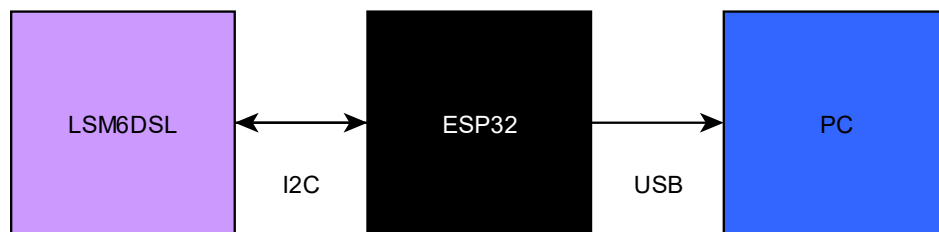


Ilustración 6: diagrama de bloques del HW.

Montaje.

El ESP32 es montado directamente sobre la lavadora. Así el eje Z está alineado con la vertical del aparato, el eje X con el lateral y el eje Y con el frontal.



Ilustración 7: Montaje del ESP32 y acelerómetro sobre lavadora.

Antes de realizar la medida durante el lavado se guardan durante unos pocos segundos los valores del acelerómetro sin ningún tipo de vibración para tener una referencia del ruido base y el valor de medida de 1g.

Posteriormente con un Script de Python se procesan los datos y se calcula la media del módulo, así el valor base de 1g es 16491.292 (1.006 g), dado que se desvía en el tercer punto decimal se toma con un valor suficientemente bueno para no tener que realizar ningún otro filtrado o ajuste de los datos.

Análisis de los datos.

Una vez se han guardado los 925684 datos estos se vuelven a cargar para su análisis. Primeramente, Representado todos los valores para un primer análisis de los datos.

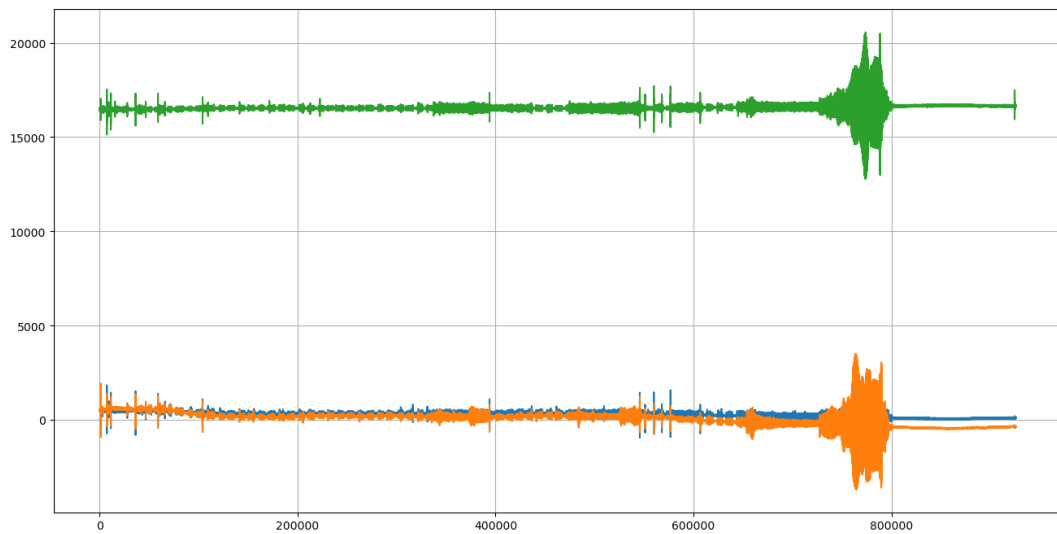


Ilustración 8: datos de vibración obtenidos durante el ensayo.

Para su análisis en profundidad, se calcula su módulo y a este se le resta el valor base de 1g. De esta manera se consigue eliminar el offset de 1g, así nos quedamos solo con las vibraciones debidas al movimiento de la propia lavadora.

En base a la gráfica del módulo se detectan diferentes zonas de lavado que serán las que se analicen. De 724000 (minuto 29) a 800000 (32 minutos) parece ser la zona de centrifugado ya que se corresponde con la de mayor intensidad que es cuando la lavadora gira con mayor velocidad. Los picos al inicio, medio (entorno a 600000) y al final en el eje Z, se corresponden con pasos cuando se pasaba cerca de la lavadora.

Así las zonas a analizar serán cerca de la muestra 800000 y cerca de la muestra 200000, se corresponden con las zonas de lavado y centrifugado.

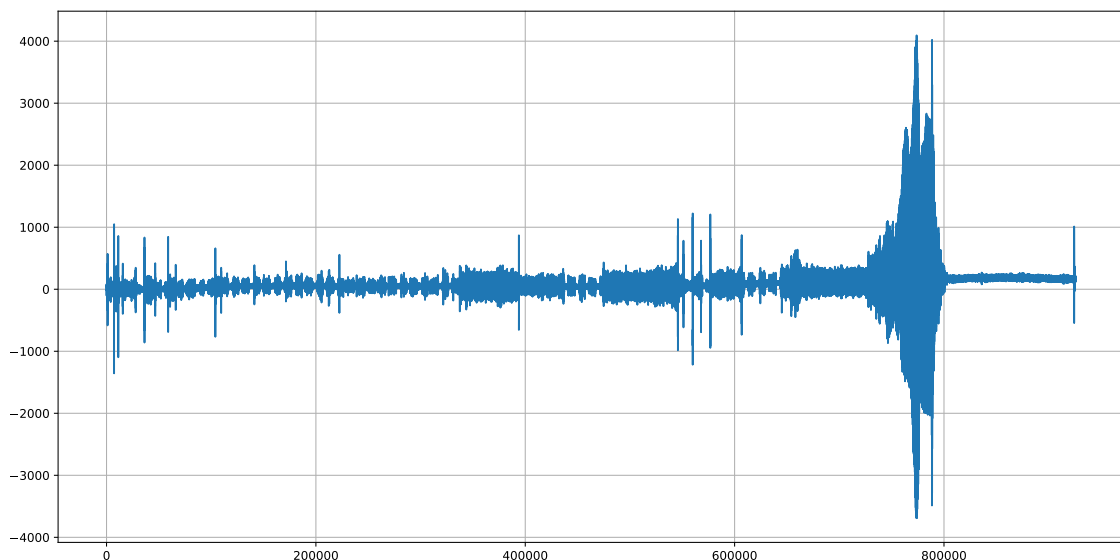


Ilustración 9: módulo de la aceleración del lavado sin el offset de la gravedad.

Para la fase de lavado se cogen los datos de la muestra 175000 y 225000, del minuto 7 al minuto nuevo. Es decir, una venta de dos minutos.

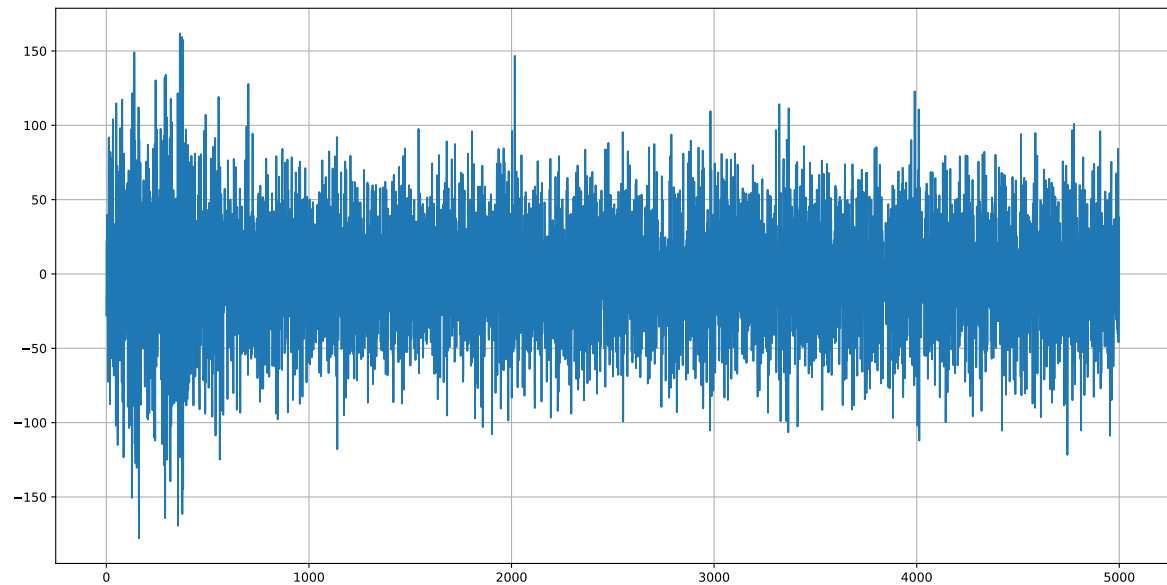


Ilustración 10: ciclo de lavado.

Después de esto se calcula su transformada de Fourier con una FFT, para ver la respuesta en frecuencia de este.

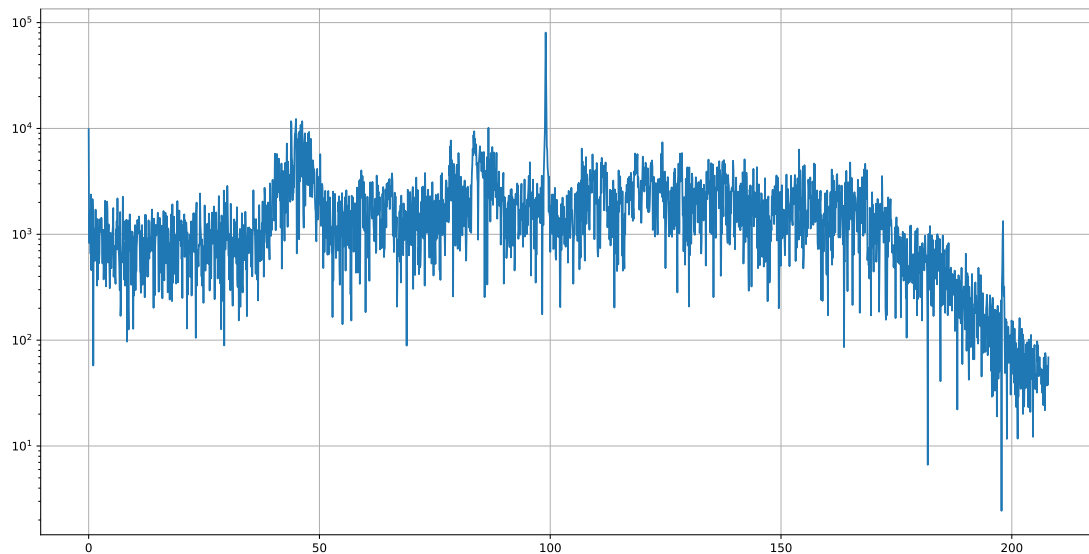


Ilustración 11: respuesta en frecuencia de la aceleración durante el lavado.

Como podemos ver existe un pico entorno a 50Hz, un poco menos con bastante ancho de banda, un pico importante en 100Hz, y bastante ruido a su alrededor. Así es muy posible que el tambor de la lavadora gire a 100 Hz. Aunque como no se puede medir la velocidad no se puede comprobar el valor.

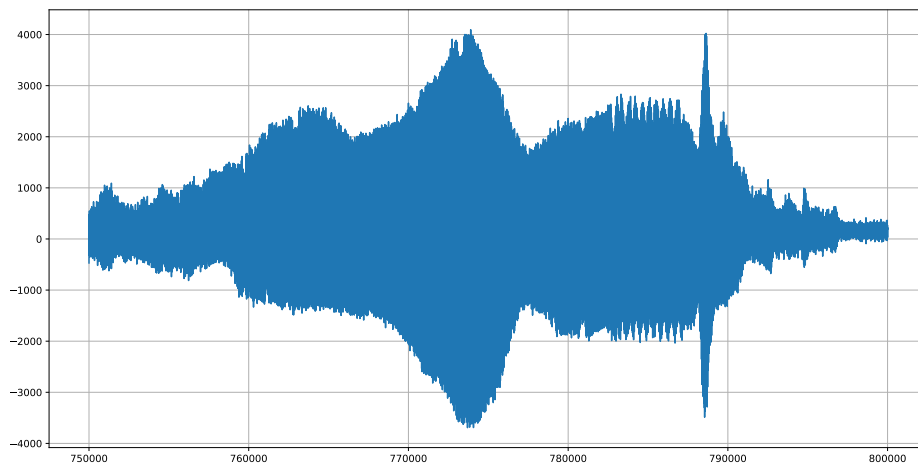


Ilustración 12: Ciclo de centrifugado.

Como se puede ver durante el ciclo de centrifugado la vibración es mucho más intensa además de presentar una respuesta más compleja. Esto es posiblemente debido a la alta velocidad de giro. Se toman las muestras de 750000 a 800000, del minuto 30 al minuto 32.

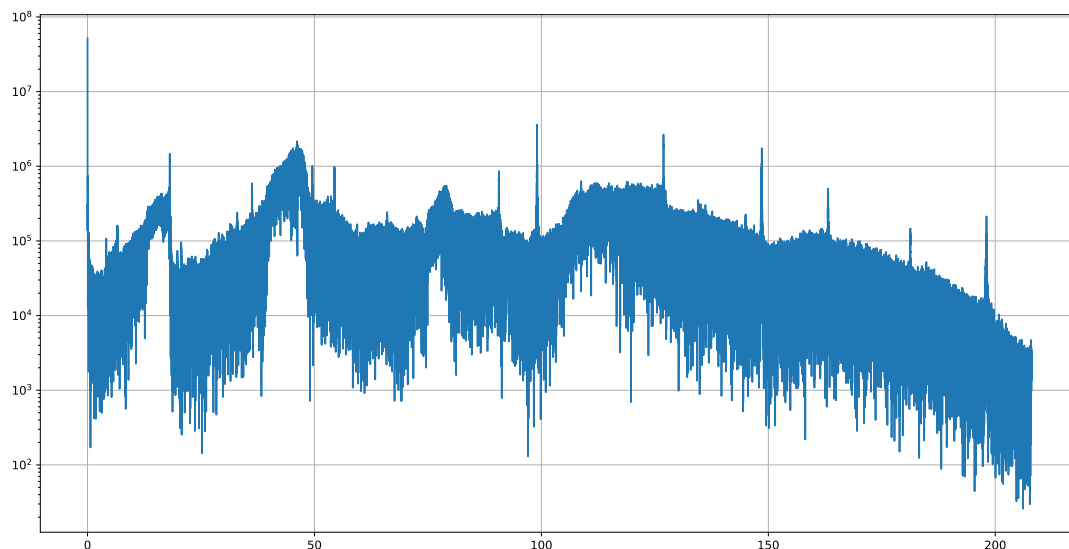


Ilustración 13: respuesta en frecuencia durante el ciclo de centrifugado.

En este caso la señal presenta un mayor ruido debido, mayor nivel en los armónicos. En este caso también se presenta un pico principal entorno a 50Hz.

Conclusiones

En conclusión, con un sencillo módulo de medida de aceleración y un montaje no muy complejo es posible detectar llegar a determinar el modo de funcionamiento de una lavadora en base a su respuesta en frecuencia.

Típicamente este tipo de análisis solo se aplica en entornos industriales, pero también es posible su aplicación en la industrial de los bienes blancos como método de comprobación del funcionamiento del aparato.

Aunque en principio la frecuencia de muestreo de 416 Hz parecía baja para la detección del modo de funcionamiento finalmente ha sido suficiente para su detección. Aunque una frecuencia superior de muestreo sería necesaria para la detección de armónicos superiores que podrían contener información relevante para detección de fallos.