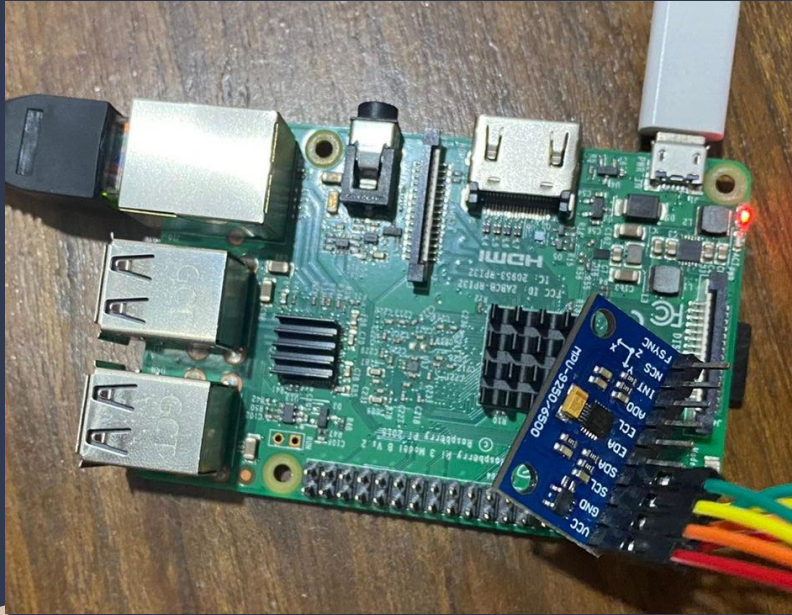


Driver I2C para sensor MPU6500

Trabajo final de la materia Implementación de
manejadores de dispositivos

Descripción



- Utilizando una Raspberry pi 3 se desarrolla un driver de linux para manejar por i2C el sensor MPU6500
- Se puede ejecutar en el SO desarrollado en la materia anterior o utilizando cualquier sistema de linux corriendo en una rpi.
- Se realizaron las prácticas de la materia para concluir con el driver I2C.
- Se desarrollaron códigos de prueba que permiten leer y escribir los datos del sensor a modo de prueba del sistema.

¿Cómo generar el driver?

Driver MPU6500

Primero se debe compilar el driver ejecutando en la consola de las raspberry:

```
rpisesteva@raspberrypi:~/Documents/IMD/mpu6500_i2c $ make
```

En caso de compilarse sin errores se debe insertar el módulo creado ejecutando:

```
rpisesteva@raspberrypi:~/Documents/IMD/mpu6500_i2c $ sudo insmod mpu6500_i2c.ko
```

Esto inicializa el módulo, para verificar que no hubo errores se pueden ver los mensajes ejecutando

```
rpisesteva@raspberrypi:~/Documents/IMD/mpu6500_i2c $ dmesg
[17704.765523] [mpu6500]: Major = 238 Minor = 0
[17704.884742] [mpu6500]: Probed!!!
[17704.884993] [mpu6500]: Driver Added!!!
```

Código del driver

```
/****** Driver functions *****/
static int mpu_open(struct inode *inode, struct file *file);
static int mpu_release(struct inode *inode, struct file *file);
static ssize_t mpu_read(struct file *filp,
|         |         |         |         char __user *buf, size_t len, loff_t * off);
static ssize_t mpu_write(struct file *filp,
|         |         |         |         const char *buf, size_t len, loff_t * off);
static long mpu_ioctl(struct file *, unsigned int, unsigned long);
/******/
static void mpu6500_write(unsigned char reg, unsigned char data);
static int mpu_i2c_read(unsigned char *out_buf, unsigned int len);
static int mpu_i2c_write(unsigned char *buf, unsigned int len);
static int mpu_i2c_read_block(const char reg, char *buff);
static int mpu6500_accelXYZ_show(int axis);
//File operation structure
```

```

int main(void)
{
    int my_dev = open("/dev/mpu_device",0);
    int data,ret,cont;
    float value;
    cont = 0;
    while(1){
        if (my_dev < 0) {
            perror("Fail to open device file: /dev/mpu_device.");
        } else {
            ret= ioctl(my_dev, 0, &data);
            value = ((float) data)/ACELLSCALE;
            printf("Se extrae la medición en x %.4f\n",value);
            usleep(STOP);
            ret= ioctl(my_dev, 1, &data);
            value = ((float) data)/ACELLSCALE;
            printf("Se extrae la medición en y %.4f\n",value);
            usleep(STOP);
            ret= ioctl(my_dev, 3, &data);
            value = ((float) data)/ACELLSCALE;
            printf("Se extrae la medición en z %.4f\n",value);
            usleep(STOP);
            ret= ioctl(my_dev, 4, &data);
            value = (float)data/TEMPSCALE + TEMPOFFSET;
            printf("Se extrae la medición de temperatura %.3f\n",value);
            usleep(STOP);
        }
        if(cont++>NUM_CYCLE) break;
    }
    close(my_dev);
    return 0;
}

```

Programas de prueba

Programas de prueba

```
int main(void)
{
    int my_dev = open("/dev/mpu_device",0);
    int data,ret,cont;
    float value;
    cont = 0;
    while(1){
        if (my_dev < 0) {
            perror("Fail to open device file: /dev/mpu_device.");
        } else {
            ret= ioctl(my_dev, 0, &data);
            value = ((float) data)/ACELLSCALE;
            printf("Se extrae la medición en x %.4f\n",value);
            usleep(STOP);
            ret= ioctl(my_dev, 1, &data);
            value = ((float) data)/ACELLSCALE;
            printf("Se extrae la medición en y %.4f\n",value);
            usleep(STOP);
            ret= ioctl(my_dev, 3, &data);
            value = ((float) data)/ACELLSCALE;
            printf("Se extrae la medición en z %.4f\n",value);
            usleep(STOP);
            ret= ioctl(my_dev, 4, &data);
            value = (float)data/TEMPSCALE + TEMPOFFSET;
            printf("Se extrae la medición de temperatura %.3f\n",value);
            usleep(STOP);
        }
        if(cont++>NUM_CYCLE) break;
    }
    close(my_dev);
    return 0;
}
```

```
#define NUM_CYCLE 20
#define STOP 200000
#define ACELLSCALE 16384
#define TEMPSCALE 340
#define TEMPOFFSET 36.53

int main(void)
{
    int my_dev = open("/dev/mpu_device",0);
    int gain,ret;
    gain = 0;
    if (my_dev < 0) {
        perror("Fail to open device file: /dev/mpu_device.");
    } else {
        ret= write(my_dev,NULL,gain);
        printf("Se graba la ganancia %d\n",gain);
        usleep(STOP);
    }
    close(my_dev);
    return 0;
}
```

Sistema funcionando

```
rpisesteva@raspberrypi:~/Documents/IMD/mpu6500_i2c $ sudo insmod mpu6500_i2c.ko  
rpisesteva@raspberrypi:~/Documents/IMD/mpu6500_i2c $ sudo ./ioctl_test.out
```

```
Se extrae la medición en x 0.0024  
Se extrae la medición en y 0.0073  
Se extrae la medición en z 0.0027  
Se extrae la medición de temperatura 37.224  
Se extrae la medición en x 0.0078  
Se extrae la medición en y 0.0007  
Se extrae la medición en z 0.0023  
Se extrae la medición de temperatura 36.954  
Se extrae la medición en x 0.0023  
Se extrae la medición en y 0.0023  
Se extrae la medición en z 0.0023  
Se extrae la medición de temperatura 36.642  
Se extrae la medición en x 0.0023  
Se extrae la medición en y 0.0010  
Se extrae la medición en z 0.0023  
Se extrae la medición de temperatura 36.618  
Se extrae la medición en x 0.0023  
Se extrae la medición en y 0.0040  
Se extrae la medición en z 0.0073  
Se extrae la medición de temperatura 36.559  
Se extrae la medición en x 0.0006  
Se extrae la medición en y 0.0021  
Se extrae la medición en z 0.0078  
Se extrae la medición de temperatura 37.036  
Se extrae la medición en x 0.0013  
Se extrae la medición en y 0.0017  
Se extrae la medición en z 0.0021  
Se extrae la medición de temperatura 37.183  
Se extrae la medición en x 0.0113  
Se extrae la medición en y 0.0098  
Se extrae la medición en z 0.0101  
Se extrae la medición de temperatura 37.280  
Se extrae la medición en x 0.0100  
Se extrae la medición en y 0.0132  
Se extrae la medición en z 0.0100  
Se extrae la medición de temperatura 36.065
```

```
Se extrae la medición de temperatura 36.071
```

```
rpisesteva@raspberrypi:~/Documents/IMD/mpu6500_i2c $ sudo ./write_test.out  
Se graba la ganancia 0
```

Ver el sistema funcionando

¿Preguntas?