FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Implementación de manejadores de dispositivos

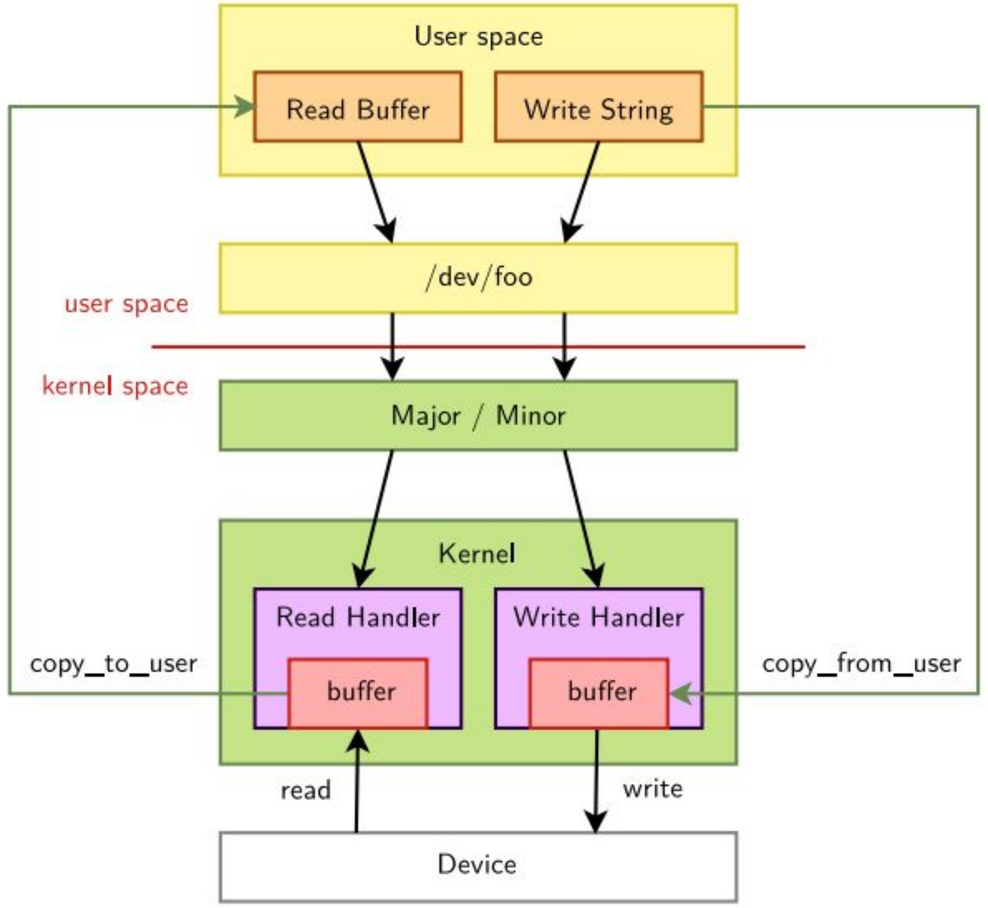
Maestría en Sistemas Embebidos

Manejo de periférico I2C con Beaglebone black

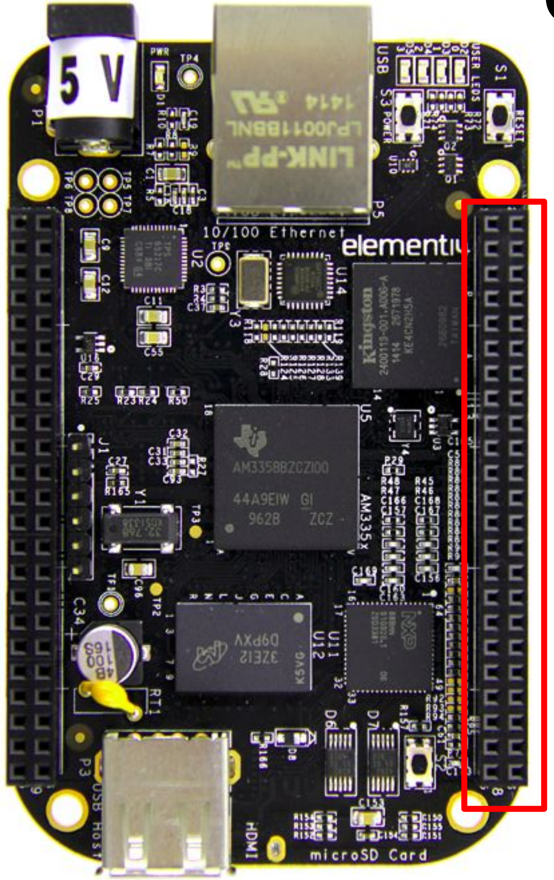
Esp. Ing. Lucas Dórdolo

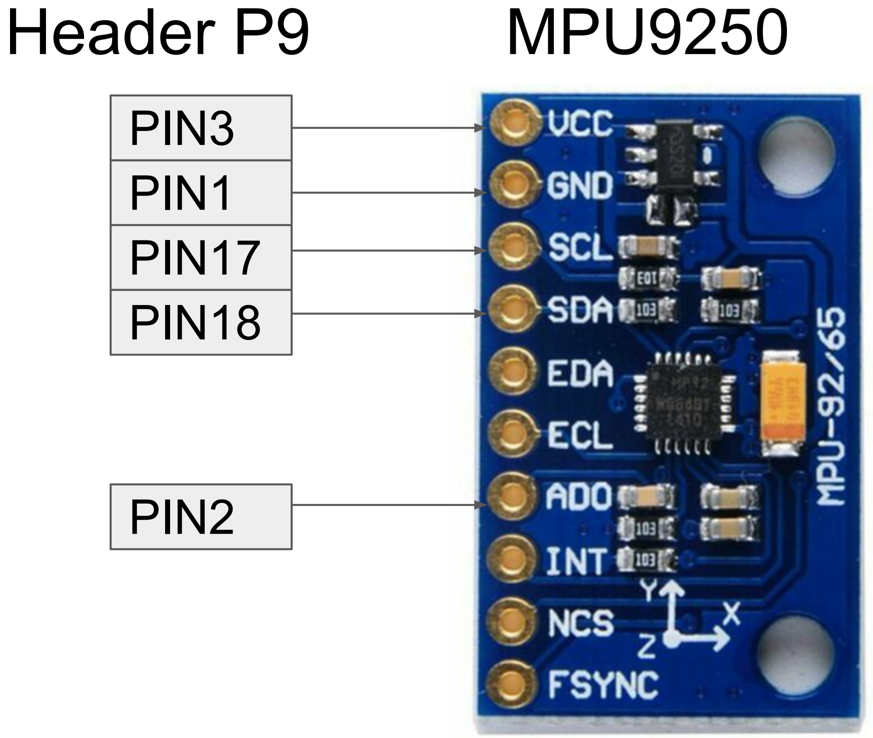
2019

Manejo de periférico: Interfaz de hardware

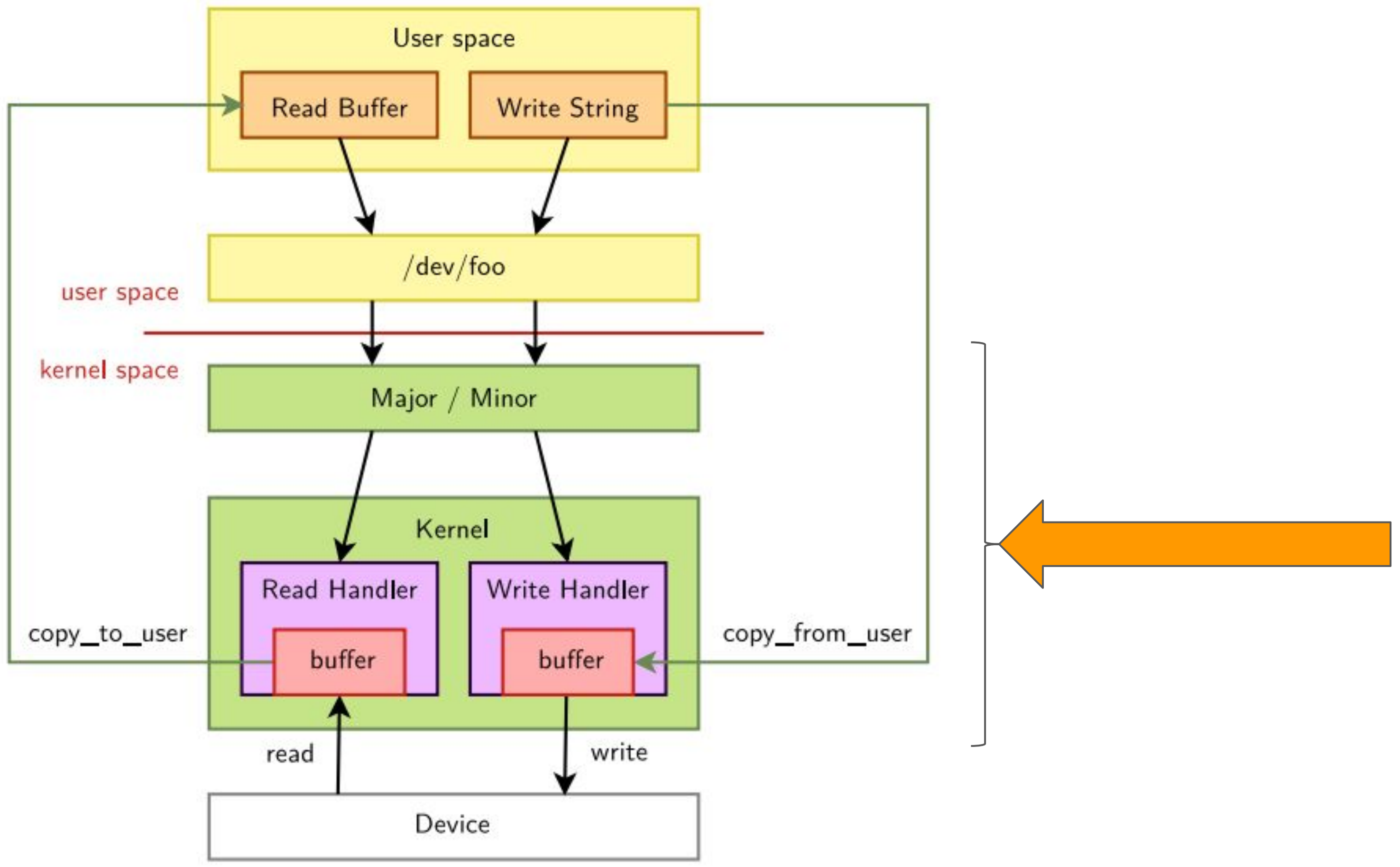


Conexión por I2C





Manejo de periférico: Módulo de kernel



Device tree: am335x-boneblack.dts

```
&am33xx_pinmux {
    i2c1_pins: pinmux_i2c1_pins {
        pinctrl-single,pins = <
            AM33XX_IOPAD(0x958, PIN_INPUT_PULLUP |
MUX_MODE2) /* spi0_d1.i2c1_sda */
            AM33XX_IOPAD(0x95c, PIN_INPUT_PULLUP |
MUX_MODE2) /* spi0_cs0.i2c1_scl */
        >;
    };
};
```

Device tree: am335x-boneblack.dts

```
&i2c1 {
    status = "okay";
    pinctrl-names = "default";
    clock-frequency = <400000>;
    pinctrl-0 = <&i2c1_pins>;

    my_mpu9250: my_mpu9250@68 {
        compatible = "mse,my_mpu9250";
        reg = <0x68>;
    };
};
```

Módulo de kernel: mympu9250.c

```
static ssize_t dev_read(struct file *filep, char *buffer, size_t len, loff_t *offset){

      int error_count = 0;
      int Ret;
      pr_info("Leyendo registros de MPU9250\n");

      Ret = i2c_master_recv(modClient, message, len);
      error_count = copy_to_user(buffer, message, len);


  ...
}
```

Módulo de kernel: mympu9250.c

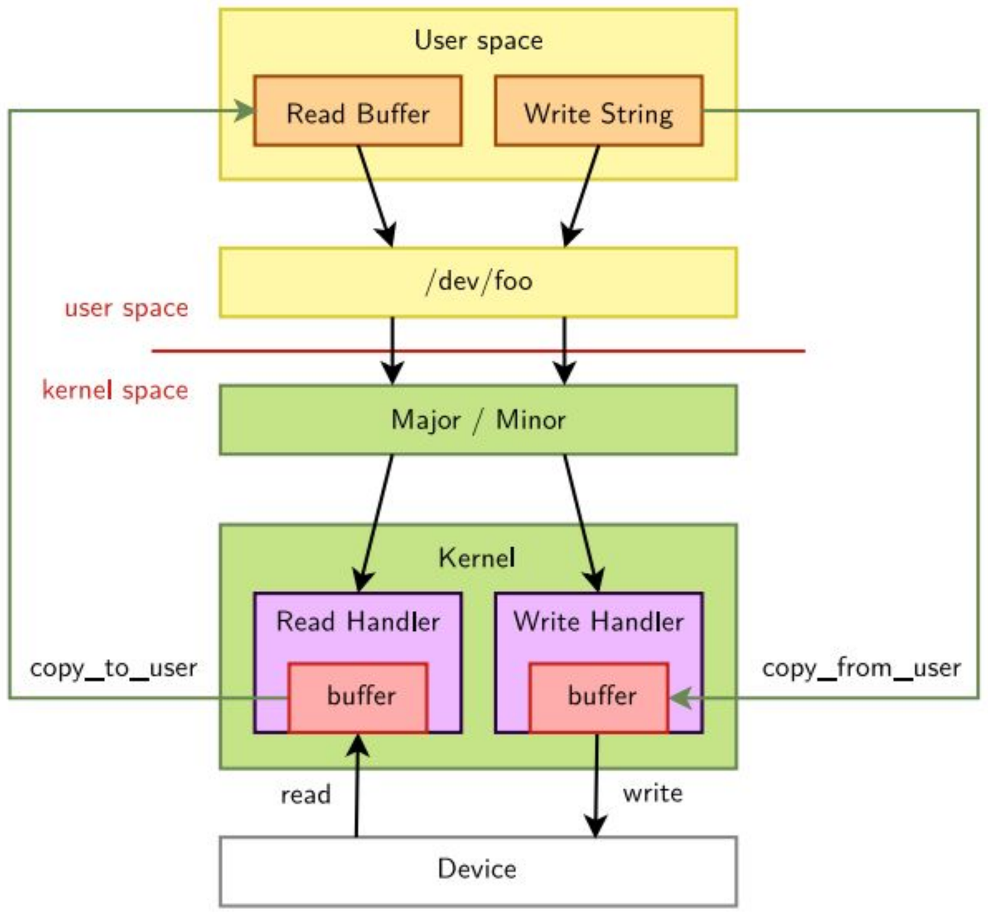
```
static ssize_t dev_write(struct file *filep, const char *buffer, size_t len, loff_t
*offset){
       int error_count = 0;

       error_count = copy_from_user(message ,buffer, len);
       error_count = i2c_master_send(modClient,message,len);

   return len;

}
```

Manejo de periférico: espacio de usuario



Aplicación de usuario: test.c

```
static int8_t mpu9250WriteRegister( uint8_t subAddress, uint8_t
data )
{
     int8_t ret = 0;
     uint8_t Buffer[2];
     Buffer[0] = subAddress; Buffer[1] = data;
     ret = write(fd, Buffer, 2);

     …
 }
```

Aplicación de usuario: test.c

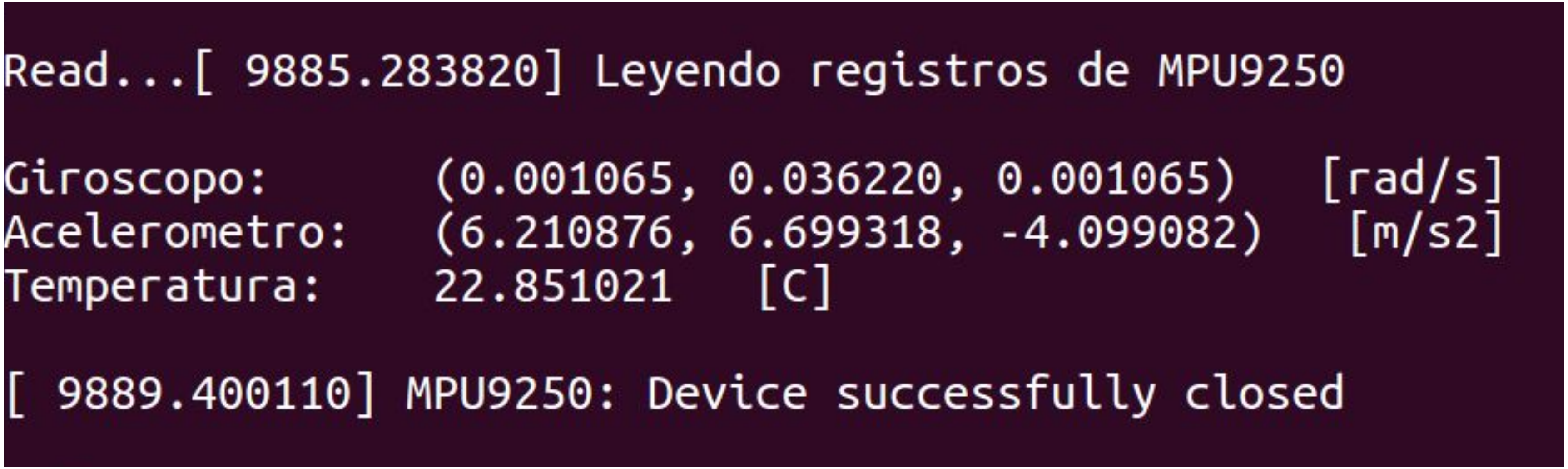
```
static int8_t mpu9250ReadRegisters( uint8_t subAddress, uint8_t
count )
{
      int ret = -1;
      uint8_t transmitDataBuffer[1] = {subAddress};
      write(fd, transmitDataBuffer, 1);
      printf("Read...\n");
    ret = read(fd, control._buffer, count);

      return 0;
}
```

Aplicación de usuario: test.c

```
int main(){
   fd = open("/dev/i2c_mse", O_RDWR);
    mpu9250Init( MPU9250_ADDRESS_0 );
    while(1){
       mpu9250Read();
               printf( "Giroscopo:   (%f, %f, %f)   [rad/s]\r\n", ...
                  );
         usleep(5000000);
    }
   printf("End of the program\n");
   return 0;
}
```

Salida en consola



```
Read...[ 9885.283820] Leyendo registros de MPU9250

Giroscopo:      (0.001065, 0.036220, 0.001065)   [rad/s]
Acelerometro:   (6.210876, 6.699318, -4.099082)   [m/s2]
Temperatura:    22.851021   [C]

[ 9889.400110] MPU9250: Device successfully closed
```

¿Preguntas?

¡Gracias!