

```

1  /*!
2  * @brief This API reads the data from the given register address
3  * of sensor.
4  */
5  int8_t bmi160_get_regs(uint8_t reg_addr, uint8_t *data, uint16_t len, const struct
bmi160_dev *dev)
6  {
7      int8_t rslt = BMI160_OK;
8
9      /* Variable to define temporary length */
10     uint16_t temp_len = len + dev->dummy_byte;
11
12     /* Variable to define temporary buffer */
13     uint8_t temp_buf[temp_len];
14
15     /* Variable to define loop */
16     uint16_t indx = 0;
17
18     /* Null-pointer check */
19     if ((dev == NULL) || (dev->read == NULL))
20     {
21         rslt = BMI160_E_NULL_PTR;
22     }
23     else if (len == 0)
24     {
25         rslt = BMI160_READ_WRITE LENGHT_INVALID;
26     }
27     else
28     {
29         /* Configuring reg_addr for SPI Interface */
30         if (dev->interface == BMI160_SPI_INTF)
31         {
32             reg_addr = (reg_addr | BMI160_SPI_RD_MASK);
33         }
34         rslt = dev->read(dev->id, reg_addr, temp_buf, temp_len);
35
36         if (rslt == BMI160_OK)
37         {
38             /* Read the data from the position next to dummy byte */
39             while (indx < len)
40             {
41                 data[indx] = temp_buf[indx];
42                 indx++;
43             }
44         }
45         else
46         {
47             rslt = BMI160_E_COM_FAIL;
48         }
49     }
50
51     return rslt;
52 }

```