**char MLX90614\_ReadTobj1(char \*pBuffer)**

**{**

**int timeOut = 0;**

**int NumByteToRead = 2;**

**while (I2C\_GetFlagStatus(I2C\_FLAG\_BUSBUSY))**

**{**

**timeOut++;**

**if (timeOut > 1000)**

**{**

**return 0;**

**}**

**}**

**I2C\_GenerateSTART(ENABLE);**

**timeOut = 0;**

**while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_MODE\_SELECT))**

**{**

**timeOut++;**

**if (timeOut > 1000)**

**{**

**return 0;**

**}**

**}**

**I2C\_Send7bitAddress(0xB4, I2C\_DIRECTION\_TX);**

**timeOut = 0;**

**while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_TRANSMITTER\_MODE\_SELECTED))**

**{**

**timeOut++;**

**if (timeOut > 1000)**

**{**

**return 0;**

**}**

**}**

**I2C\_ClearFlag(I2C\_FLAG\_ADDRESSSENTMATCHED);**

**I2C\_SendData(0x07); /\* MSB \*/**

**timeOut = 0;**

**while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_BYTE\_TRANSMITTED))**

**{**

**timeOut++;**

**if (timeOut > 1000)**

**{**

**return 0;**

**}**

**}**

**I2C\_GenerateSTART(ENABLE);**

**timeOut = 0;**

**while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_MODE\_SELECT))**

**{**

**timeOut++;**

**if (timeOut > 1000)**

**{**

**return 0;**

**}**

**}**

**I2C\_Send7bitAddress(0xB4, I2C\_DIRECTION\_RX);**

**timeOut = 0;**

**while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_RECEIVER\_MODE\_SELECTED))**

**{**

**timeOut++;**

**if (timeOut > 1000)**

**{**

**return 0;**

**}**

**}**

**I2C\_ClearFlag(I2C\_FLAG\_ADDRESSSENTMATCHED);**

**timeOut = 0;**

**while (NumByteToRead)**

**{**

**if (NumByteToRead == 1)**

**{**

**I2C\_AcknowledgeConfig(I2C\_ACK\_NONE);**

**I2C\_GenerateSTOP(ENABLE);**

**}**

**if (I2C\_CheckEvent(I2C\_EVENT\_MASTER\_BYTE\_RECEIVED))**

**{**

**\*pBuffer = I2C\_ReceiveData();**

**pBuffer++;**

**NumByteToRead--;**

**}**

**{**

**timeOut++;**

**if (timeOut > 1000)**

**{**

**return 0;**

**}**

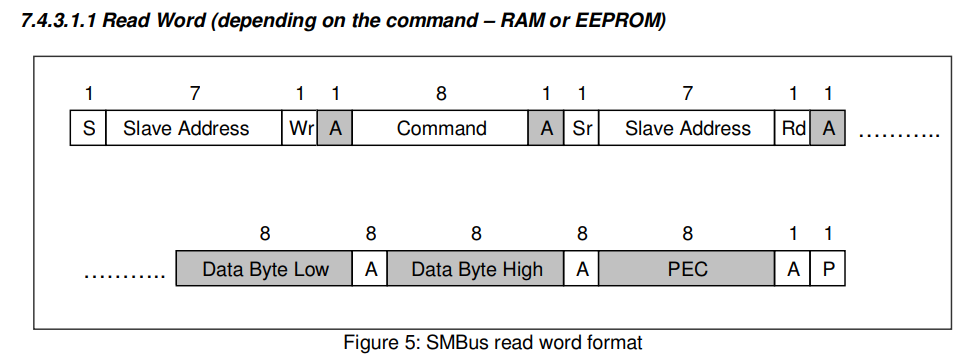
**}**

**}**

**I2C\_AcknowledgeConfig(I2C\_ACK\_CURR);**

**return 1;**

**Cách đọc dữ liệu MLX90614**

****

char MLX90614\_ReadTobj1(char \**pBuffer*)

{

  int timeOut = 0; *// nếu giao tiếp lỗi thì bị timeOut và chương trình tiếp tục hoạt động, không bị treo*

  int NumByteToRead = 2; *// đọc 2 byte data từ MLX90614*

  while (I2C\_GetFlagStatus(I2C\_FLAG\_BUSBUSY)) *// kiểm tra line i2c xem sẵn sàng không*

  {

    timeOut++;

    if (timeOut > 1000)

    {

      return 0;

    }

  }

  I2C\_GenerateSTART(ENABLE); *// gửi tín hiệu băt đầu giao tiếp*

  timeOut = 0;

  while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_MODE\_SELECT)) *// kiểm tra xem đã chọn được mode master chưa*

  {

    timeOut++;

    if (timeOut > 1000)

    {

      return 0;

    }

  }

  I2C\_Send7bitAddress(0xB4, I2C\_DIRECTION\_TX);  *// gửi địa chỉ slave, yêu cầu ghi*

  timeOut = 0;

  while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_TRANSMITTER\_MODE\_SELECTED)) *// đợi tín hiệu của slave*

  {

    timeOut++;

    if (timeOut > 1000)

    {

      return 0;

    }

  }

  I2C\_ClearFlag(I2C\_FLAG\_ADDRESSSENTMATCHED); *// xoá cờ*

  I2C\_SendData(0x07); *// gửi bản tin địa chỉ ram cần đọc (nhiệt độ Tobj)*

  timeOut = 0;

  while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_BYTE\_TRANSMITTED)) *// đợi salve phản hồi*

  {

    timeOut++;

    if (timeOut > 1000)

    {

      return 0;

    }

  }

  I2C\_GenerateSTART(ENABLE); *// gửi tín hiệu repeated start cho slave*

  timeOut = 0;

  while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_MODE\_SELECT)) *// kiểm tra xem đã chọn mode master chưa*

  {

    timeOut++;

    if (timeOut > 1000)

    {

      return 0;

    }

  }

  I2C\_Send7bitAddress(0xB4, I2C\_DIRECTION\_RX);  *// gửi địa chỉ slave, yêu cầu đọc*

  timeOut = 0;

  while (!I2C\_CheckEvent(I2C\_EVENT\_MASTER\_RECEIVER\_MODE\_SELECTED)) *// đợi slave trả lời*

  {

    timeOut++;

    if (timeOut > 1000)

    {

      return 0;

    }

  }

  I2C\_ClearFlag(I2C\_FLAG\_ADDRESSSENTMATCHED); *// xoá cờ*

  timeOut = 0;

  while (NumByteToRead) *// đợi đến khi đọc đủ 2 byte*

  {

    if (NumByteToRead == 1) *// nếu chỉ còn phải nhận 1 byte*

    {

      I2C\_AcknowledgeConfig(I2C\_ACK\_NONE); *//bỏ tín hiệu ack sau khi nhận byte cuối cùng*

      I2C\_GenerateSTOP(ENABLE); *// gửi tín hiệu stop ngừng giao tiếp*

    }

    if (I2C\_CheckEvent(I2C\_EVENT\_MASTER\_BYTE\_RECEIVED))   *// kiểm tra nếu có dữ liệu đến*

    {

      \**pBuffer* = I2C\_ReceiveData(); *// đọc dữ liệu*

*pBuffer*++;

      NumByteToRead--;

    }

    {

      timeOut++;

      if (timeOut > 1000)

      {

        return 0;

      }

    }

  }

  I2C\_AcknowledgeConfig(I2C\_ACK\_CURR); *// đưa chế độ ack về gửi ack sau khi nhận dữ liệu để giao tiếp lần sau*

  return 1;

}