

- Read accelerometer data
- Read gyroscope data
- Read temperature
- Read everything above in one single function or separatelly
- Use 2 independent MPU-6050 devices on the same I2C
 - Devices must have different ADO pin state

Dependencies

- CMSIS
 - STM32F4xx
 - STM32F4xx RCC
 - STM32F4xx GPIO
 - STM32F4xx I2C
- TM
 - TM I2C

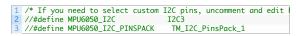


defines.h



MPU-6050	STM32F4	Description
SCL	PA8	I2C clock line
SDA	PC9	I2C data line
ADO		LSB address pin. If pin is high, address is 0xD2, otherwise 0xD0
VCC		3.3V power supply
GND		GND

By default, I2C3 is used for communication. If you want to change it, open defines.h file and edit lines:



Note: If you want to connect 2 MPU6050 devices, you have to connect them to the same I2C pins but with different ADO pin on MPU6050 device.

Functions and enumerations

```
* MPU6050 can have 2 different slave addresses, depends on
* This feature allows you to use 2 different sensors with
                          ADD pin is set to low
TM_MPU6050_Device_0:
ADD pin is set to low
TM_MPU6050_Device_1:
ADD pin is set to high
 10
typedef enum {
12     TM_MPU6050_Device_0 = 0,
13     TM_MPU6050_Device_1 = 0x02
14     } TM_MPU6050_Device_t;
          /**

* Result enumeration

*
18
19
20
21
22
            * ParameterS:
* - TM_MPU6050_Result_0k:

- TM_MPU6050_Result_DeviceNotConnected:
There is no device with valid slave address
- TM_MPU6050_Result_DeviceInvalid:
```

```
iii January 2016
December 2015
iii November 2015
Cotober 2015
August 2015
iii March 2015
iii February 2015
iii January 2015
December 2014
Ctober 2014
ä August 2014
iii July 2014
iii lune 2014
```

```
Connected device with address is not MPU6050
          Typedef enum {
   TM_MPU6050_Result_Ok = 0x00,
   TM_MPU6050_Result_DeviceNotConnected,
   TM_MPU6050_Result_DeviceInvalid
} TM_MPU6050_Result_t;
 27
28
           /**
 * Set parameters for accelerometer range
 34
35
 36
 37
38
39
                             - TM_MPU6050_Accelerometer_2G:
Range is +- 2G
                              - TM_MPU6050_Accelerometer_4G:
Range is +- 4G
- TM_MPU6050_Accelerometer_8G:
Range is +- 8G
- TM_MPU6050_Accelerometer_16G:
 42
43
 44
45
                                          Range is +- 16G
 46
                      edet enum {
TM_MPU6050_Accelerometer_2G = 0×00,
TM_MPU6050_Accelerometer_4G = 0×01,
TM_MPU6050_Accelerometer_8G = 0×02,
TM_MPU6050_Accelerometer_16G = 0×03
 47
48
49
 50
51
52
53
54
55
           } TM_MPU6050_Accelerometer_t;
            /**
 * Set parameters for gyroscope range
 56
57
58
59
                             - TM_MPU6050_Gyroscope_250s:
Range is +- 250°/s
                              Range is +- 2507/s

- TM_MPU6050_Gyroscope_500s:
Range is +- 500°/s

- TM_MPU6050_Gyroscope_1000s:
Range is +- 1000°/s

- TM_MPU6050_Gyroscope_2000s:
Range is +- 20000°/s
 64
65
            typedef enum {
TM_MPU6050_Gyroscope_250s = 0x00,
TM_MPU6050_Gyroscope_500s = 0x01,
 66
67
68
69
                       TM_MPU6050_Gyroscope_1000s = 0x02,
TM_MPU6050_Gyroscope_2000s = 0x03
           } TM_MPU6050_Gyroscope_t;
            /**
 * Main MPU6050 struct
               * Parameters:
* - uint8_t Address:
 76
77
78
79
80
                               I2C address of device
Only for private use
- float Gyro_Mult:
                              - Floot Gyro_Mult:
Gyroscope corrector from raw data to "degrees/s'
Only for private use
- float Acce_Mult:
Accelerometer corrector from raw data to "g"
Only for private use
 81
82
 83
84
85
86
87
88
89
90
91
                               - int16_t Accelerometer_X:
    Accelerometer value X axis
- int16_t Accelerometer_Y:
    Accelerometer value Y axis
- int16_t Accelerometer_Z:
    Accelerometer value Z axis
                              Accelerometer Value 2 (
int16_t Gyroscope_X:
Gyroscope value X axis
- int16_t Gyroscope_Y:
Gyroscope value Y axis
- int16_t Gyroscope_Z:
 93
94
95
96
97
98
99
100
                               Gyroscope value Z axis - float Temperature:
                                          Temperature in degrees
 101
102
            typedef struct {
   /* Private */
   uint8_t Address;
   float Gyro_Mult;
   float Acce_Mult;
   /* Public */
   inthe companies
 105
106
107
          /* Public */
int16_t Accelerometer_X;
int16_t Accelerometer_Y;
int16_t Accelerometer_Z;
int16_t Gyroscope_X;
int16_t Gyroscope_Y;
int16_t Gyroscope_Z;
float Temperature;
} TM_MPU6050_t;
 108
109
110
117
118 /**
119 * Initialize MPU6050
120 *
 120
121
122
              * Parameters:

* - TM_MPU6050_t* DataStruct:
                                - TM_MPU6050_t* DataStruct:
Pointer to TM_MPU6050_t empty working struct
- TM_MPU6050_Device_t DeviceNumber:
Device number. MPU6050 has one pin, AD0 which ca
This feature allows you to use 2 different senso
If you set AD0 pin to low, then this parameter s
but if AD0 pin is high, then you should use TM_MI
- TM_MPU6050_Accelerometer_t AccelerometerSensitivity
Set accelerometer sensitivity
- TM_MPU6050_Gyroscope_t GyroscopeSensitivity:
Set gyroscope sensitivity
 124
 128
 131
132
              * Returns TM_MPU6050_Result_Ok if everything is OK */
 136 extern TM_MPU6050_Result_t TM_MPU6050_Init(TM_MPU6050_t* Da
138 /**
```

Privacy & Cookies Policy

```
* Read accelerometer data from sensor
140
             - TM_MPU6050_t* DataStruct:
                Pointer to TM_MPU6050_t struct to store accelera
     * Returns TM_MPU6050_Result_Ok if everything is OK
    extern TM_MPU6050_Result_t TM_MPU6050_ReadAccelerometer(TM_
148
149
150
   /**

* Read gyroscope data from sensor
151
152
153
     * Parameters:
            - TM_MPU6050_t* DataStruct:
Pointer to TM_MPU6050_t struct to store gyroscope
156
157
     * Returns TM_MPU6050_Result_Ok if everything is OK
158 extern TM_MPU6050_Result_t TM_MPU6050_ReadGyroscope(TM_MPU6
160 /**
161
162
     * Read temperature data from sensor
            - TM_MPU6050_t* DataStruct:
               Pointer to TM_MPU6050_t struct to store temperate
167 * Returns TM_MPU6050_Result_0k if everything is OK
168 */
169 extern TM_MPU6050_Result_t TM_MPU6050_ReadTemperature(TM_MP
170
171
172
    /**
* Read accelerometer, gyrscope and temperature data from se
     * Parameters:
            - TM MPU6050 t* DataStruct:
                 Pointer to TM_MPU6050_t struct to store data
     * Returns TM_MPU6050_Result_Ok if everything is OK
180 extern TM_MPU6050_Result_t TM_MPU6050_ReadAll(TM_MPU6050_t*
```

Example 1

- One MPU-6050 device is connected to pins PA8 and PC9 on STM32F4
- AD0 of MPU6050 is connected to GND, gives us address 0xD0
- Read is performed every 500ms, displayed on USART1 (PB6, at 115200 baud) to the computer

```
Keil project example for 2 MPU6050 6-axes sensors
                   Before you start, select your target, on the right of
 6 *
                                               tilen@majerle.eu
                    @email
                   @website http://stm32f4-discovery.net
@ide Keil uVision 5
                   @packs
@stdperiph
                                              STM32F4xx Keil packs version 2.2.0 or g
STM32F4xx Standard peripheral drivers ve
10 *
* @stdperiph STM32F4xx Si

*/ Include core modules */

#include "stm32f4xx.h"

/* Include my libraries here */

#include "defines.h"

#include "tm_stm32f4_delay.h"

#include "tm_stm32f4_usart.h"

#include "tm_stm32f4_mpu6050.h"
       #include <stdio.h>
        int main(void) {
   TM_MPU6050_t MPU6050_Data0;
   TM_MPU6050_t MPU6050_Data1;
   uint8_t sensor1 = 0, sensor2 = 0;
                 char str[120];
                    * Initialize system */
30
31
32
33
                SystemInit();
                   * Initialize delay */
                TM_DELAY_Init();
34
35
36
37
               /* Initialize USART, TX: PB6 */
TM_USART_Init(USART1, TM_USART_PinsPack_2, 115200);
                /* Initialize MPU6050 sensor 0, address = 0xD0, AD0 pin if (TM_MPU6050_Init(&MPU6050_Data0, TM_MPU6050_Device_0
40
41
42
43
44
45
46
47
                        /* Display message to user */
TM_USART_Puts(USART1, "MPU6050 sensor 0 is ready to
                       /* Sensor 1 OK */
sensor1 = 1;
               /* Initialize MPU6050 sensor 1, address = 0xD2, AD0 pin
if (TM_MPU6050_Init(&MPU6050_Data1, TM_MPU6050_Device_1
    /* Display message to user */
    TM_USART_Puts(USART1, "MPU6050 sensor 1 is ready to
                        /* Sensor 2 0K */
sensor2 = 1;
```

```
while (1) {
                                 if (TM_DELAY_Time() >= 500)
   /* Reset time */
   TM_DELAY_SetTime(0);
58 59
60
61
62
63
64
65
                                                 * If sensor 1 is connected */
                                             if (sensor1) {
   /* Read all data from sensor 1 */
   TM_MPU6050_ReadAll(&MPU6050_Data0);
                                                     /* Format data */
sprintf(str, "1. Accelerometer\n- X:%d\n-
MPU6050_Data0.Accelerometer_X,
MPU6050_Data0.Accelerometer_Y,
MPU6050_Data0.Accelerometer_Z,
MPU6050_Data0.Gyroscope_X,
MPU6050_Data0.Gyroscope_Y,
MPU6050_Data0.Gyroscope_Z,
MPU6050_Data0.Temperature
68 69
70
71
72
73
                                                       /* Show to usart */
TM_USART_Puts(USART1, str);
80
81
82
83
                                                   If sensor 2 is connected */
                                                   (sensor2) {
  /* Read all data from sensor 1 */
  TM_MPU6050_ReadAll(&MPU6050_Data1);
                                                       /* Format data */
sprintf(str, "2. Accelerometer\n- X:%d\n-
MPU6050_Data1.Accelerometer_X,
88
89
                                                                 MPU6050_Data1.Accelerometer_Y,
MPU6050_Data1.Accelerometer_Z,
MPU6050_Data1.Gyroscope_X,
MPU6050_Data1.Gyroscope_Y,
MPU6050_Data1.Gyroscope_Z,
MPU6050_Data1.Temperature
90
91
92
93
                                                       /* Show to usart */
TM_USART_Puts(USART1, str);
```

Example 2

- Two MPU-6050 devices are connected to STM32F4, both to pins PA8 and PC9
- First device has ADO pin connected to GND, gives us **0xDO**
- Second device has ADO pin connected to 3.3V, gives is 0xD2 address
- Both devices are send to USART

```
Keil project example for 2 MPU6050 6-axes sensors
                      Before you start, select your target, on the right of
                      @author
                                                    Tilen Majerle
                     @author | Illen Majerte
@email tilen@majerle.eu
@website http://stm32f4-discovery.net
@ide Keil uVision 5
@packs STM32F4xx Keil packs version 2.2.0 or g
@stdperiph STM32F4xx Standard peripheral drivers v
 10
11
12
11 * @stdperiph STM32F4xx St
12 */
13 /* Include core modules */
14 #include "stm32f4xx.h"
15 /* Include my libraries here */
16 #include "defines.h"
17 #include "tm_stm32f4_delay.h"
18 #include "tm_stm32f4_usart.h"
19 #include "tm_stm32f4_mpu6050.h"
 20
21
22
        #include <stdio.h>
        int main(void) {
   TM_MPU6050_t MPU6050_Data0;
   TM_MPU6050_t MPU6050_Data1;
   uint8_t sensor1 = 0, sensor2 = 0;
   character[110].
26
27
28
                 char str[120];
29
30
31
32
                     * Initialize system */
                 SystemInit():
                    * Initialize delay */
                 TM_DELAY_Init();
                 /* Initialize USART, TX: PB6 */
TM_USART_Init(USART1, TM_USART_PinsPack_2, 115200);
                 /* Initialize MPU6050 sensor 0, address = 0xD0, AD0 pin if (TM_MPU6050_Init(&MPU6050_Data0, TM_MPU6050_Device_0
38
39
40
41
42
43
                          /* Display message to user */
TM_USART_Puts(USART1, "MPU6050 sensor 0 is ready to
                           /* Sensor 1 OK */
```

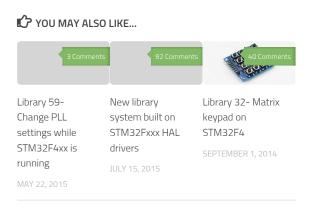
Privacy & Cookies Policy

```
46
47
                    /* Initialize MPU6050 sensor 1, address = 0xD2, AD0 pin
if (TM_MPU6050_Init(&MPU6050_Data1, TM_MPU6050_Device_1
    /* Display message to user */
    TM_USART_Puts(USART1, "MPU6050 sensor 1 is ready to
 48
49
 50
51
52
53
54
55
                             /* Sensor 2 OK */
sensor2 = 1;
 56
57
58
59
60
61
62
63
64
65
                     while (1) {
                                      Every 500ms */
                              // if (TM_DELAY_Time() >= 500000) {
   /* Reset time */
   TM_DELAY_SetTime(0);
                                         /* If sensor 1 is connected */
                                       if (sensor1) {
   /* Read all data from sensor 1 */
   TM_MPU6050_ReadAll(&MPU6050_Data0);
 66
67
68
69
                                                /* Format data */
sprintf(str, "1. Accelerometer\n- X:%d\n-
MPU6050_Data0.Accelerometer_X,
MPU6050_Data0.Accelerometer_Y,
MPU6050_Data0.Accelerometer_Z,
MPU6050_Data0.Gyroscope_X,
MPU6050_Data0.Gyroscope_Y,
MPU6050_Data0.Gyroscope_Z,
MPU6050_Data0.Temperature
70
71
72
73
74
75
76
77
 78
79
80
81
82
83
                                                 /* Show to usart */
TM_USART_Puts(USART1, str);
                                         /* If sensor 2 is connected */
                                        if (sensor2) {
    /* Read all data from sensor 1
 84
85
86
87
88
89
                                                 TM MPU6050 ReadAll(&MPU6050 Data1):
                                                 /* Format data */
sprintf(str, "2. Accelerometer\n- X:%d\n-
MPU6050_Data1.Accelerometer_X,
 90
91
92
93
94
                                                          MPU6050_Data1.Accelerometer_Y, MPU6050_Data1.Accelerometer_Z,
                                                          MPU6050_Data1.Gyroscope_X,
MPU6050_Data1.Gyroscope_Y,
MPU6050_Data1.Gyroscope_Z,
                                                           MPU6050_Data1.Temperature
                                                /* Show to usart */
TM_USART_Puts(USART1, str);
 100
101
102
103 }
```

Both projects are available on Github, download library below.

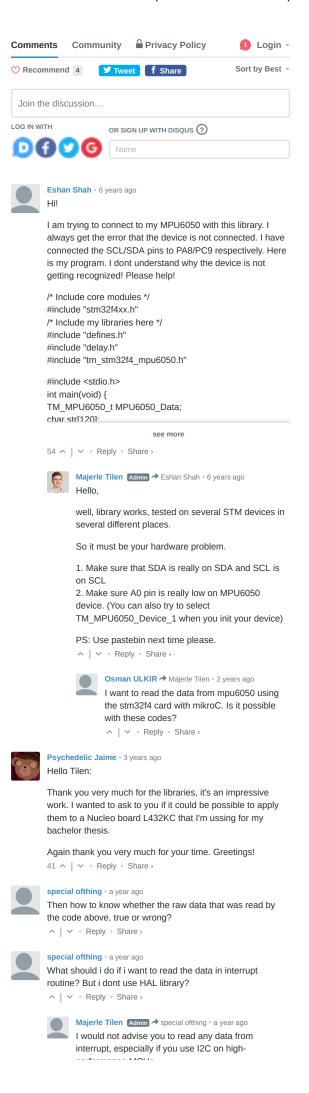


Privacy & Cookies Policy



Privacy & Cookies Policy

Privacy & Cookies Policy



About LinkedIn Twitter Google+ Facebook Github Instagram performance MCUs. It takes a lot of time to read. I would prefer setting a flag in the interrupt and then read data in main function. A | V • Reply • Share > special ofthing → Majerle Tilen • a year ago how about i use timer interrupt to read any data? Is it the same case with i2c interrupt routine? So you would not advise it? ^ | ∨ 1 • Reply • Share > Majerle Tilen Admin → special ofthing • a year ago I do not recomment reading this from interrupt, unless you only start DMA or interrupt-based reading from interrupt. If you need precise timebase for reading, use FIFO feature of the sensor and define precise sampling rate, then read with bursts from main pediredla naresh • 2 years ago I am doing I2C project with [MPU6050][1] using [STM32F4DISCOVERY][2] board. The ADDR bit in SR1 register is clearing in one instruction execution without reading SR2 register. Because of clearing ADDR bit, I can't do acknowledge disable is made during EV6 in single byte receiving case. The datasheet states: >ADDR: Address sent (master mode)/matched (slave mode) This bit is cleared by software reading SR1 register followed reading SR2. In case a single byte has to be received, the Acknowledge disable is made during EV6 (before ADDR flag is cleared) and the STOP condition generation is made after EV6. If I write any instruction (or declaration) between `I2C->DR=address|0` and `while(!(I2C2->SR1 & (1<<1)))` in see more A | V • Reply • Share > Salvador Apablaza • 3 years ago Hi Majerle, first of all, thanks a lot for your libraries, amazing I'm having several issues trying to replicate this examples with SW, mostly with includes, defines, linker instructions, etc, so i decided the best way to do it would be to use the Privacy & Cookies Policy same ide you do, i think its keil, isnt it?

interface? Been looking for a way to visualize the output, but so far only got serial windows for UARTs.

Salvador.

^ | ∨ • Reply • Share >

Thanks for your time Majerle,



Rakesh Rajbhar • 4 years ago Hello Majerle Tilen,

thanks for the very effective library

i want to develop the application for step count of human so for that i am using MPU6050. and i have developed code for MPU6050 with stm32f100 series microcontroller

i am able to get correct WhoAml regster value,i have configured all the registers as you have done.

when i am trying to read raw data from the MPU6050 using TM_MPU6050_ReadAll(&MPU6050_Data) function. my device is stable on the table but i am getting this type of data from MPU6050,i am not getting below data.

and i want to convert the data into Gauss And "degrees/s" so how can i get this result please help me.

see more

Privacy & Cookies Policy

11 of 13 19/05/21, 14:42

12 of 13 19/05/21, 14:42

19/05/21, 14:42 13 of 13