

Biblioteca MPU para a Caixa Preta

Versão 1.0, 16/04/2020

Funções

void	mpu_rd_ac_gi_mg	(word *vetor)
void	mpu_rd_mg	(word *vetor)
void	mpu_mag_config	(void)
void	mpu_int	(void)
void	mpu_int	(void)
void	mpu_rd_ac_tp_gi	(word *vetor)
void	mpu_rd_ac_gi	(word *vetor)
void	mpu_acorda	(void)
void	mpu_dorme	(void)
byte	mpu_whoami	(void)
void	mpu_inicializa	(void)
void	mpu_escalas	(byte gfs, byte afs)
void	void mpu_sample_rt	(byte sample_rate)
void	mpu_wr	(byte reg, byte dado)
byte	mpu_rd	(byte reg)
void	mpu_wr_blk	(byte reg, byte *dado, byte qtd)
void	mpu_rd_blk	(byte reg, byte *dado, byte qtd)

- void **mpu_rd_ac_gi_mg** (word *vetor)
Ler Aceleração, Giro e Mag. Retorna vetor de 18 bytes. Não monta as palavras de 16 bits.
Retorna: [axh axl ayh ayl azh azl gxh gxl gyh gyl gzh gzl hxx hxl hyh hyl hzh hzl].
- void **mpu_rd_mg** (word *vetor)
Ler Magnetômetro. Retorna vetor de 3 words [hx hy hz].
- void **mpu_mag_config** (void)
Inicializar Magnetômetro – Ainda não sabemos como fazer
- void **mpu_int** (void)
Preparar para MPU usar INT4
Pino PE4 entrada com pullup e habilitar INT4 para flanco de descida
MPU: interrupção em baixo com push-pull, pulso de 50 useg
MPU: Habilitar interrupção dado pronto
- void **mpu_des_int** (void)
Desabilitar o uso de interrupção pelo MPU.
Desabilitar INT4 e MPU: Desabilitar interrupção dado pronto.
- void **mpu_rd_ac_tp_gi** (word *vetor)
Ler Aceleração, temperatura e giro. Retorna vetor de 7 words [ax ay az tp gx gy gz].
- void **mpu_rd_ac_gi** (word *vetor)
Ler Aceleração e giro. Retorna vetor de 6 words [ax ay az gx gy gz].

- void **mpu_acorda** (void)
Acordar o MPU e programar para usar relógio Giro X.
- void **mpu_dorme** (void)
Dormir o MPU e programar para usar relógio Giro X.
- byte **mpu_whoami** (void)
Retornar a leitura do registrador WHO_AM_I.
- void **mpu_config** (void)
Colocar o MPU num estado conhecido.
Taxa = 1 kHz, Banda: Acel=5 Hz e Giro=5 Hz e delay=19 mseg
Taxa de amostragem = taxa/(1+SMPLRT_DIV) = 1k/10 = 100Hz
Escalas acel = +/-2g e giro = +/-250 gr/s
- void **mpu_escalas** (byte gfs, byte afs)
Selecionar Fundo de Escalas para o MPU.
Acel: 0=+/-2g, 1=+/-4g, 2=+/-8g, 3=+/-16g.
Gyro: 0=+/-250gr/s, 1=+/-500gr/s, 2=+/-1000gr/s, 3=+/-2000gr/s.

```
#define GIRO_FS_250  0    // +/- 250 graus/seg
#define GIRO_FS_500  1    // +/- 500 graus/seg
#define GIRO_FS_1000 2    // +/- 1000 graus/seg
#define GIRO_FS_2000 3    // +/- 2000 graus/seg

#define ACCEL_FS_2G   0    // +/- 2g
#define ACCEL_FS_4G   1    // +/- 4g
#define ACCEL_FS_8G   2    // +/- 8g
#define ACCEL_FS_16G  3    // +/- 16g
```

- void **mpu_sample_rt** (byte sample_rate)
Selecionar Sample Rate, considerando Taxa = 1kHz (Registrador CONFIG)

```
// Valores para o Sample Rate, Registrador SMPLRT_DIV
// Considerando Taxa = 1kHz (Registrador CONFIG)
#define SAMPLE_RT_1kHz      0    // 1.000/(0+1) = 1000
#define SAMPLE_RT_500Hz     1    // 1.000/(1+1) = 500
#define SAMPLE_RT_333Hz     2    // 1.000/(2+1) = 333,33
#define SAMPLE_RT_250Hz     3    // 1.000/(3+1) = 250
#define SAMPLE_RT_200Hz     4    // 1.000/(4+1) = 200
#define SAMPLE_RT_166Hz     5    // 1.000/(5+1) = 166,66
#define SAMPLE_RT_142Hz     6    // 1.000/(6+1) = 142,85
#define SAMPLE_RT_125Hz     7    // 1.000/(7+1) = 125
#define SAMPLE_RT_111Hz     8    // 1.000/(8+1) = 111,11
#define SAMPLE_RT_100Hz     9    // 1.000/(9+1) = 100
```

- void **mpu_wr** (byte reg, byte dado)
(10) Escrever num registrador do MPU.
- byte **mpu_rd** (byte reg)
(20) Ler um registrador do MPU.
- void **mpu_wr_blk** (byte reg, byte *dado, byte qtd)
(30) Escrever um bloco de dados no MPU a partir de um registrador.

- void **mpu_rd_blk** (byte reg, byte *dado, byte qtd)
(40) Ler um bloco do MPU a partir de um registrador.