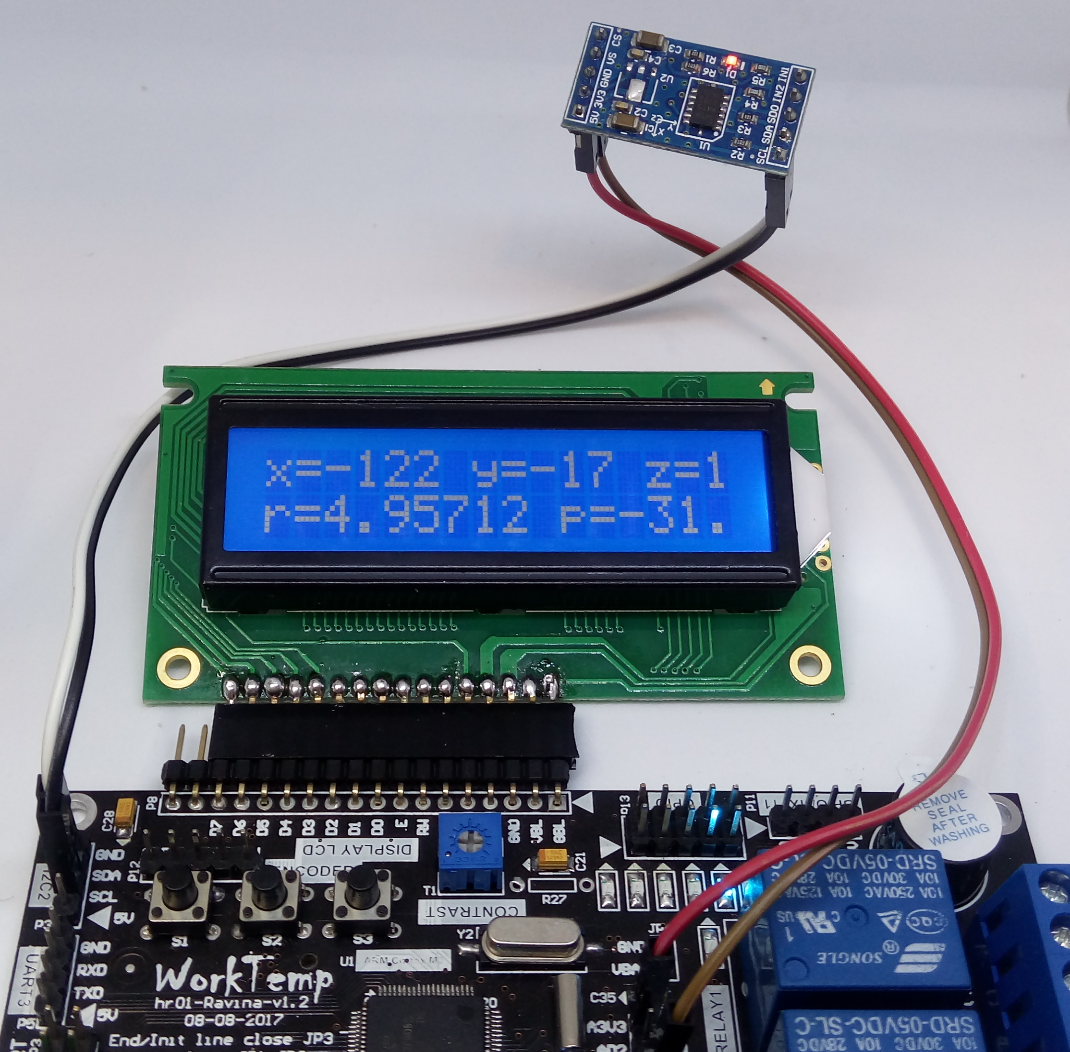
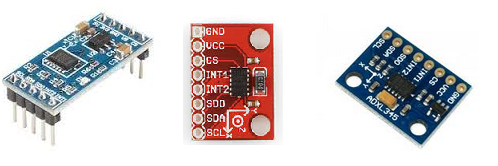
**Utilizando o Acelerômetro Digital**

Neste exemplo mostra como usar acelerômetro de três eixos modelo adxl345, ele usa uma resolução de 13 bits com medição de até ±16g. O adxl345 é adequado para medir acelerações estáticas de gravidade em aplicações sensíveis, tal como acelerações dinâmicas resultando de movimentação, impacto ou queda livre. Com sua alta resolução (4mg/LSB) é capaz de medir medições de inclinações menor que 1.0º. O adxl345 também pode identificar batidas leves seja única batida ou dupla batida em sequência, conceito chamado de TAP, um exemplo de utilizada dessas batidas seria os clicks em um mouse pad de um notebook.



**Conexão com o Kit ARM**

Geralmente estes acelerômetros são vendidos em Shields com os demais componentes necessários para sua perfeita funcionalidade, e neste exemplo estamos usando um desses módulos:

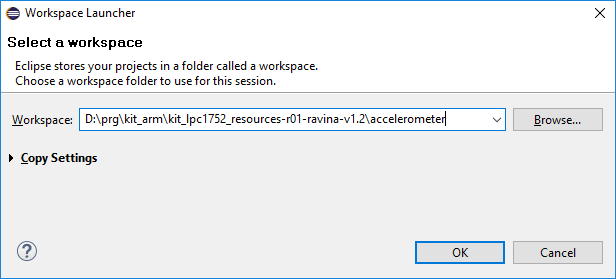


Independe de usar Shields ou não, a conexão é semelhante. Nesse shield estamos usando a porta de comunicação I2C e usando a alimentação de 3V. A tabela abaixo descreve a conexão:

|  |  |
| --- | --- |
| **Módulo Display** | **Conector** |
| SDA | P3 pino 3 |
| SCL | P3 pino 2 |
| VCC 3v3 | P9 pino 1 |
| GND | P9 pino 4 |

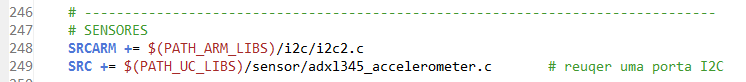
**Programa Exemplo**

Abra o exemplo da pasta “accelerometer” com o Eclipse:

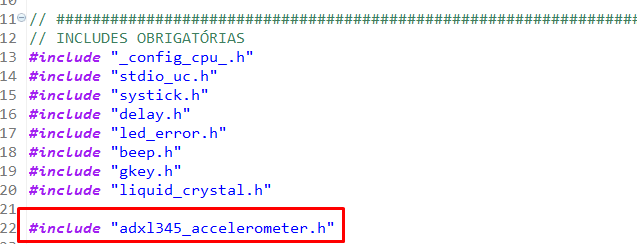


No makefile, na secção de inserção de fontes, foi incluindo as seguintes fontes:

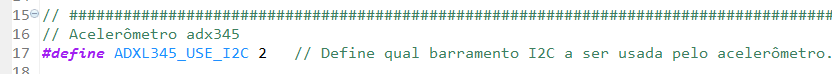
* I2c2.c: Código de acesso a porta I2C do ARM;
* adxl345\_accelerometer.c: Controlador do acelerômetro.



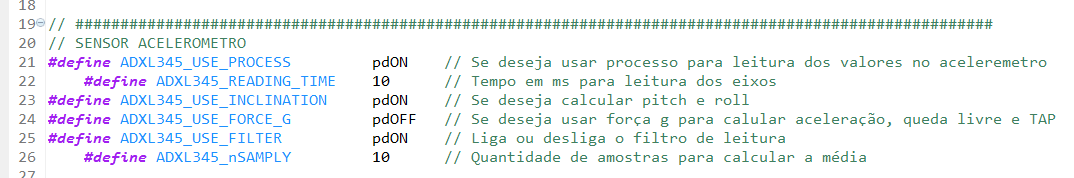
Em app.h foi adicionado a include do acelerômetro para usar as suas funções na aplicação:



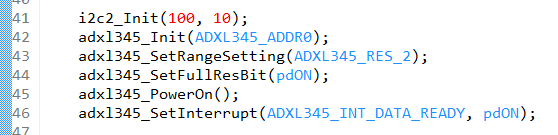
Em \_config\_cpu\_i2c.h é feito a configuração de qual porta I2C de acesso ao acelerômetro:



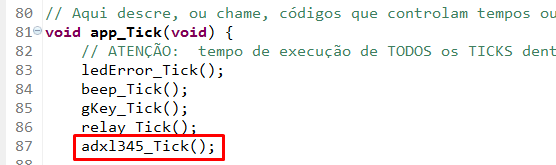
Em \_config\_lib\_sensor.h são feito as configurações de como o acelerômetro deve trabalhar e quais recursos serão fornecidos por ele:



Em main.c dentro do int main(void), é inicializado a porta I2C com velocidade de 100KHz com timeout na espera máxima pela resposta do dispositivo de 10ms, também inicializamos acelerômetro com as devidas configurações de trabalho para esta determinada aplicação:



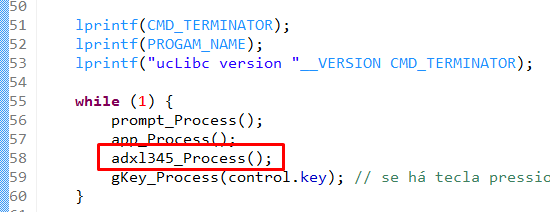
A leitura do sensor é feita em tempos regulares é preciso chamar a função adxl345\_Tick dentro da função app\_Tick. A função app\_Tick chama todas as funções ticks do sistema:



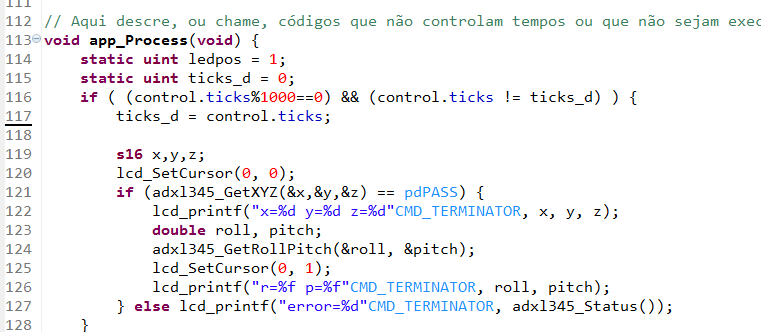
A função app\_Tick é chamada a cada 1ms, isto foi especificado no início do programa na função main:



O processo de leitura é feito de forma automática pelo driver, e para isto é preciso adicionar o adxl345\_Process dentro do loop da função main:

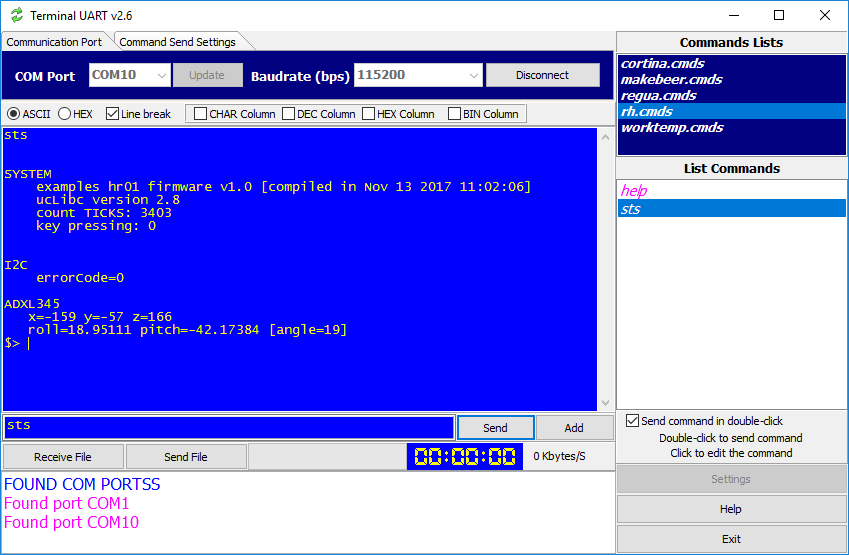


Em app\_Process mostramos os valores lidos do acelerômetro tais como os eixos x, y, z, e mostramos o roll e pich:

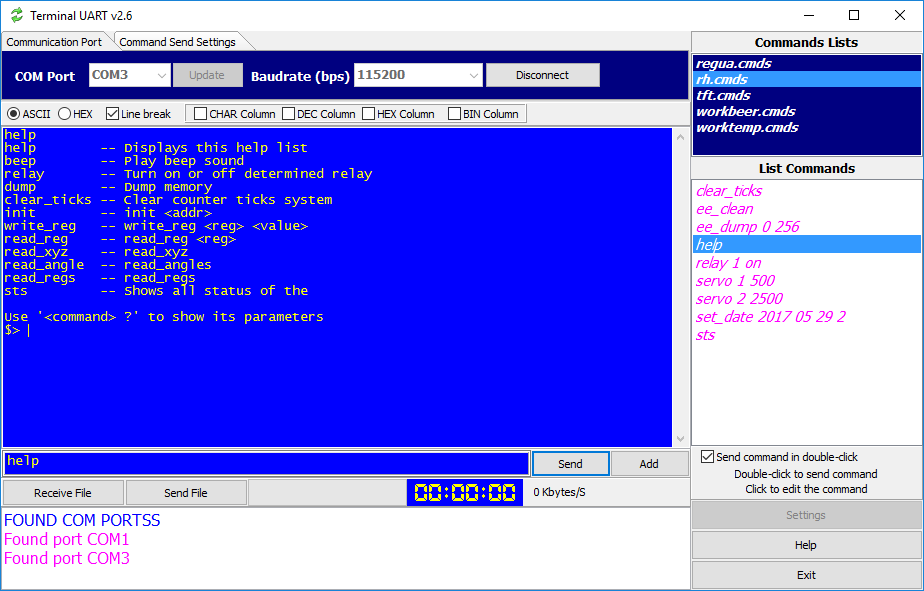


**Compile e Execute**

Com tudo instalado e configurado, compile a aplicação, grave e execute no ARM. Use o CLI junto ao um terminal para enviar comandos de leitura do acelerômetro, abaixo mostra a leitura dos eixos e ângulos de inclinação quando emitido o comando de status:



Para mais comandos digite help:



**Acelerômetro adxl345**

O adxl345 é um acelerômetro de três eixos com uma resolução de 13 bits com medição de até ±16g. A saída de dados digital é formatada em dois bytes por eixo. Os dados dos eixos e os registradores podem ser acessados tanto via interface SPI ou como I2C.

O adxl345 é adequado para medir acelerações estáticas de gravidade em aplicações sensíveis, tal como acelerações dinâmicas resultando de movimentação ou impacto. Com sua alta resolução (4mg/LSB) é capaz de medir medições de inclinações menor que 1.0º.

O adxl345 também pode identificar batidas leves seja única batida ou dupla batida em sequência, conceito chamado de TAP, um exemplo de utilizada dessas batidas seria o click e duplo click com batidas do dedo em um mouse pad de um notebook. Os tempos e duração e intervalos das batidas são configuráveis.

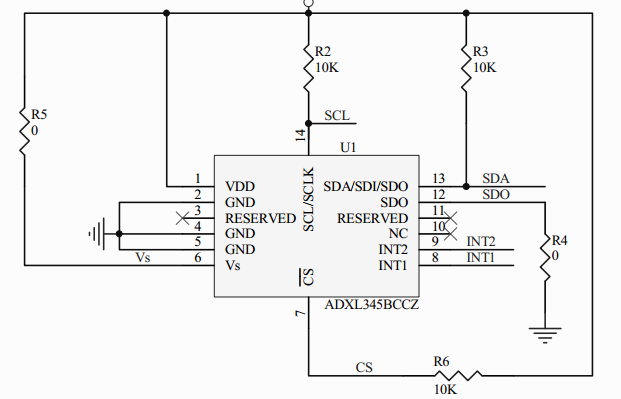
O adxl345 também pode identificar quando o mesmo sofrer queda livre.

**Ligações**

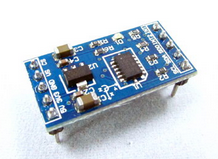
O adxl345 se comporta como escravo no barramento e pode ser acessado via SPI ou I2C. Para acessar via I2C o pino #CS deve sempre ficar em alta. Quando o pino #CS vai em baixa o componente entre em modo de comunicação SPI. Como a maioria dos Shields são preparados para usar o barramento I2C, apesar de permitir acessos via SPI, vamos programar a nossa biblioteca usando acesso via I2C. O adxl345 não há resistores de pullups internos nos seus pinos, requerendo resistores de 10K externos.

Para comunicação I2C o pino #CS deve ficar em alta e suporta duas velocidades, a velocidade padrão de 100KHz e a velocidade rápida de 400KHz. Acesso de simples e múltiplos bytes são suportados. Como o pino ALT ADDRESS em alta, os sete bits de endereço I2C do dispositivo é 0x3A, seguido pelo bit R/#W, colocando o pino em baixo o endereço se torna 0xA6.

Há dois pinos de alimentação a ser usado, o pino VS e o pino VDD. A combinações de ligações destes pinos de alimentações permite condições diferentes de trabalho referente a consumo para economizar energia, como em casso não estão trabalhando com aparelhos que utilizam baterias e não requer economia da mesma, vamos ligar ou desligar os dois pinos juntos, ou seja, coloca-los em curto circuito a alimentação:



Caso esteja usando um Shield não precisa se preocupar com os resistores pullup e nem as ligações dos pinos de configurações do adxl345, eles já vêm preparados prontos para serem usados com barramento I2C, basta ligar a alimentação e o barramento:



**Iniciando**

Após a alimentação o adxl345 fica em standby, mesmo assim os registradores podem ser acessados para leitura e escrita. Contudo, é preciso esperar no mínimo 1.1ms antes de acessar seus registradores após liga-lo. Para iniciar as medições é preciso acessar o bit3 do POWER\_CTL (addr 0x2d). **OBS:** É sempre recomendado fazer todas as configurações no modo standby antes de ligar o dispositivo para medições e as interrupções.

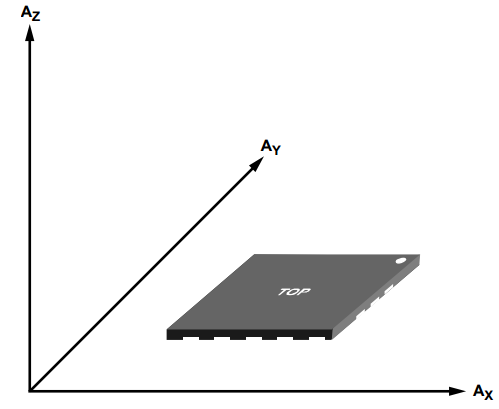
A mínima sequência para inicialização do adxl345 para fazer as medições são:

1. Ajustar o formato dos dados no registrador DATA\_FORMAT configurando a resolução (FULL\_RES) e alcance da leitura (RANGE);
2. Tirar o adxl345 do modo standby para o modo de medição. Isto é feito colocando o bit MEASURE em 1 no registrador POWER\_CTL;
3. Habilitar a interrupção quando os dados estão prontos, isto é feito colocando o bit DATA\_READY em 1 no registrador INT\_ENABLED. Mesmo que não vai utilizar o pino de interrupção.

Neste caso o adxl345 vai iniciar com as demais operações no padrão, ou seja, sem uso de FIFO, sem uso do TAP, sem detecção de queda livre, modo baixo consumo desligado e a velocidade de atualização das saídas de 100Hz.

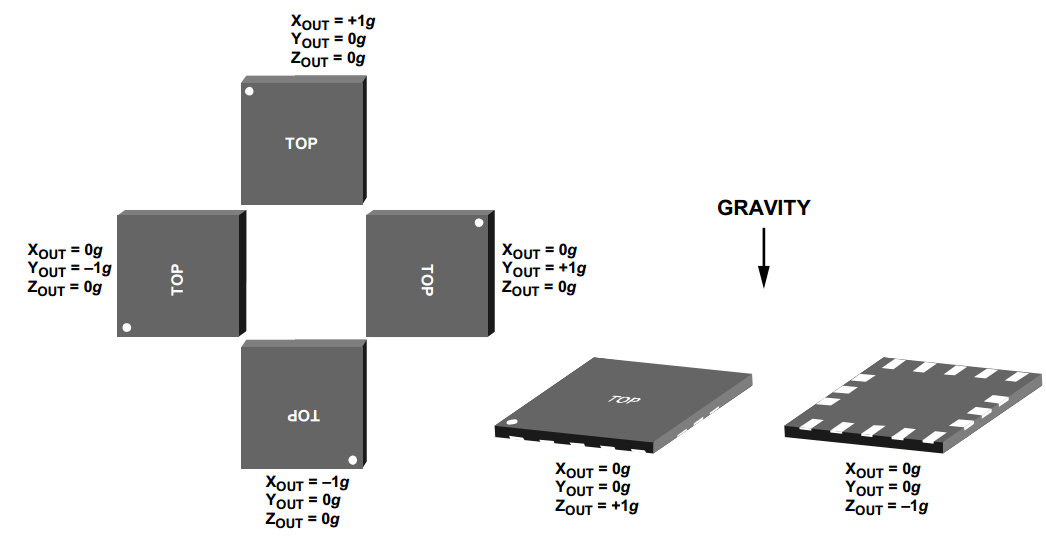
**Leitura dos Sensores**

Há três eixos de sentido de aceleração para ser lido pelo adx345, os eixos X e Y para medir a aceleração nos sentidos das laterais do adxl345, e o eixo Z para medir a aceleração no sentido para cima e para baixo do adxl345. Na figura abaixo mostra os eixos de aceleração em relação a posição física do adxl345, o nome top do desenho sinaliza a parte de cima do adxl que voltada para cima (contra a gravidade):



As detecções das acelerações são positivas, ou seja, os valores são capturados quando os sensores aceleram no sentido dos seus respectivos eixos.

A detecção da aceleração não acontece somente quando o adxl sai da inércia para um determinado movimento, também conhecida como força g, mas também detecta a força da gravidade quando o adxl está em repouso. Em um adxl em repouso e alinhado ao horizonte a influência da gravidade em seus eixos são:



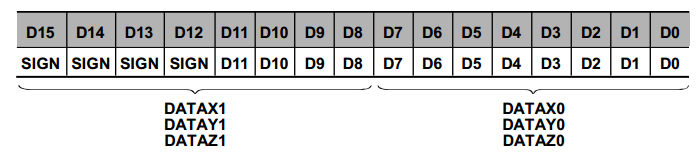
Quando o adxl345 está 90° do seu eixo Z, de lado para gravidade, os eixos que vão sofrer a aceleração são os eixos X e Y como mostrado na figura acima a esquerda, onde mostra os eixos X e Y variando entre -1g a 1g de acordo com sua posição. Quando adxl345 está deitado em um plano alinhado com o horizonte os eixos X e Y sempre serão 0 e o eixo Z varia de -1g a 1g quando girando no sentido da gravidade.

A FIFO contém 32 de 32 níveis armazenando para cada eixo e pode trabalhar de 4 modos: ByPass; FIFO; Stream e Trigger. Porém, os mais interessantes são:

* ByPass: Não é utilizado a FIFO, os registradores dos três eixos são obtidos diretos dos conversores ao invés da FIFO. Neste modo a interrupção DATA\_READY é normalmente usada sinalizando que um novo dado está pronto para ser lido;
* FIFO: Os dados dos três eixos são armazenados na FIFO. Quando o nível da FIFO é igual ao registrador FIFO\_CTL (addr 0x38) uma interrupção é emitida. Se a FIFO ficar cheia os dados novos serão descartados;
* Stream: Os dados dos três eixos são armazenados na FIFO. Quando o nível da FIFO é igual ao registrador FIFO\_CTL (addr 0x38) uma interrupção é emitida. Se a FIFO ficar cheia os dados antigos serão descartados.

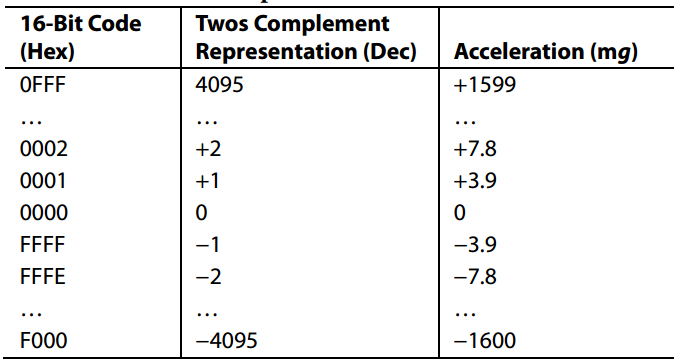
Os dados da FIFO são lidos através dos registradores DATAX, DATAY e DATAZ. Quando o modo da FIFO não está em modo ByPass, lendo os registradores DATAX, DATAY e DATAZ da FIFO estais lendo os valores mais antigos desta, dando lugar para novos dados a serem armazenado, mesmo que a metade das informações dos eixos foram lidos, ou seja, a FIFO sofre um POP. Logo, todos os três eixos devem ser lidos em um único acesso, isto além de garantir que todos os três eixos são lidos na mesma amostragem.

Os registradores DATAXYZ são de 16 bits onde os 4 bits MSB sinaliza o sinal e os demais bits os valores:



Se não há mais novos dados os registradores DATAXYZ permanecerão com os dados antigos.

Quando usado a resolução em 13bist o bit menos significativo representa 3.9mg:

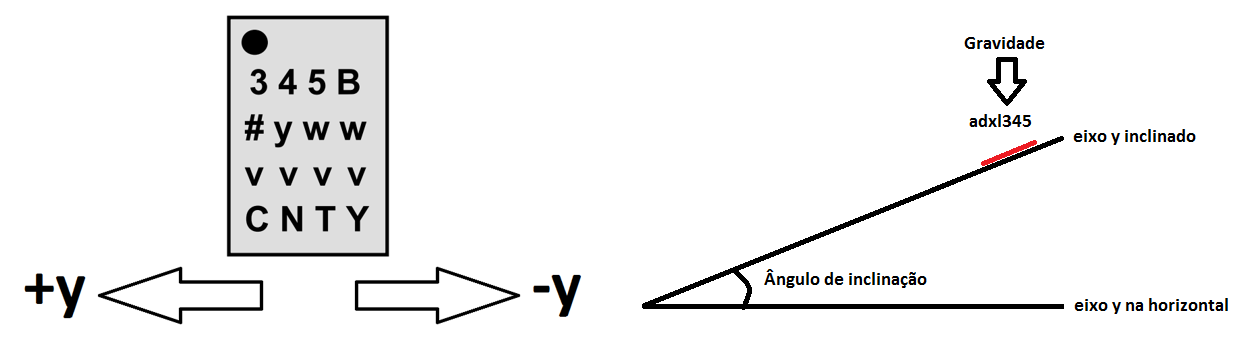


**Filtrando**

O sensor é muito sensível e detecta quaisquer vibrações aplicado a ele, se deseja ler os sensores para determinar ângulos de inclinação em um local sujeito a vibrações mecânicas, ou seja, então é preciso aplicar um filtro para descartar tais vibrações. Usar um filtro passa baixa é uma boa opção, entretanto, o mais recomendado é usar o filtro Kalman, este é mais complexo e oferece o sinal mais limpo de ruídos.

**Calculando a Inclinação Vertical**

Para calcular a inclinação vertical é preciso conhecer as forças g dos eixos z e y. Para isto o adxl345 deve ficar em uma posição onde o seu eixo Z fique perpendicular à gravidade e o eixo Y na linha de inclinação. Na figura abaixo mostra com deve ficar posicionado o adxl345:



A imagem da esquerda mostra o adxl345 vista de cima, no sentido da gravidade, e com o eixo Y no sentido da inclinação. E na imagem da direita mostra vista lateral.

Para encontrar o ângulo basta aplicar a seguinte fórmula:

**float** angle = atan2(z, y) \* 180.0f / PI

ADXL345 DataSheet Digital Accelerometer

ADXL345 Quick Start Guide - AN-1077 APPLICATION NOTE - by Tomoaki Tusuzki

<https://github.com/mandeluna/inclinometer>

<https://www.sparkfun.com/tutorials/240>

<http://blog.oscarliang.net/use-gy80-arduino-adxl345-accelerometer/>

<http://stackoverflow.com/questions/3797604/how-to-filter-accelerometer-data-from-noise>

<http://theccontinuum.com/2012/09/24/arduino-imu-pitch-roll-from-accelerometer/>