

# Relatório Projeto Washing Machine notifier

Otho Teixeira Komatsu

Hevelyn Sthepany Lima

Matrículas: 170020142 - 170059031

Professor: Edson Mintsu Hung

Disciplina: Laboratório de Sistemas Microprocessados

Turma: B

July 13, 2019



## 1 Descrição

Trata-se de um programa embarcado em **MSP430** direcionado a lavanderias ou usos domésticos, que consiste em um dispositivo de notificação acoplado a uma máquina de lavar, que avisa via email o término de uma lavagem. A detecção é baseada no tremor da máquina de lavar, fenômeno comum durante a lavagem.

Seu funcionamento é da seguinte maneira:

- **Ligado/Reiniciado:** O MSP sinaliza o início de seu funcionamento (indicado no código pelo enum **IDLE**) piscando 2 vezes simultaneamente os **LEDS vermelho e verde**. Após isso, ele aguarda o comando de acionamento.
- **Comando de acionamento:** O MSP deve ser acionado pelo pressionamento da **chave S2**, que responderá com 2 piscares do **LED vermelho**. Nessa fase, pressupõe-se que a máquina já está no processo de lavagem, já fornecendo tremor ao sensor MPU-6050. Então, o dispositivo inicia o processo de análise da ocorrência da lavagem.
- **Inicialização:** O MSP depois das piscadas do **LED vermelho** está no processo de coleta das amostragens iniciais do giroscópio proveniente do MPU-6050 (estado indicado no código pelo enum **START**). Essa etapa visa a iniciar a leitura de medidas do MPU-6050, cujos primeiros valores são medidas inválidas e instáveis, portanto não constataando na análise do estado de lavagem. Após a conclusão dessa etapa, o dispositivo piscará 2 vezes o **LED verde** indicando a transição para a etapa de amostragem efetiva de valores válidos.
- **Amostragem:** Após a inicialização, o sensor passa a verificar o funcionamento da máquina (condição do sensor indicado no código pelo enum **START**), amostrando 10 valores da velocidade angular do eixo-y do seu giroscópio a cada 1 segundo, e calculando sua média. Se seu valor estiver fora de um intervalo de valores de estabilidade(intervalo que indicaria o repouso do sensor, e portanto, da Máquina de lavar); ele estaria em processo de lavagem(indicado no código pelo enum **WASHING**). Caso contrário, ele estaria no estado de repouso, ou seja, término de lavagem (indicado no código pelo enum **FINISH**).
- **Término de lavagem:** Após o dispositivo detectar o fim da lavagem, o processo de notificação via email inicia.
- **Reset:** Caso o desejo de reinicializar todo o processo, basta pressionar o botão **RST(S3)** do MSP.

## 2 Abordagens

Para a reprodução do procedimento acima, foram criados dois **enums** que indicariam estados, tanto a do sensor quanto a do MSP. A do sensor:

- Start: Estado inicial do sensor quando começa a amostrar. Cumpre o objetivo de fazer as primeiras amostras que apresentam valores irregulares e instáveis, portanto irrelevantes na análise do programa.
- Sample: Estado em que o sensor começa a de fato fornecer valores de interesse ao MSP (no caso a velocidade angular do eixo y do giroscópio do sensor). Nesse estado, o MSP passa a analisar os valores fornecidos pelo sensor.

A do dispositivo:

- Idle: Esse estado indica que o MSP acabou de ser ligado/reiniciado, pronto para ser acionado a detectar o funcionamento da máquina de lavar.
- Washing: Esse estado é assumido logo que o dispositivo é acionado. Isso pressupõe que ao ser acionado, a máquina de lavar já está em processo de lavagem, portanto, tremendo e indicando isso ao sensor. Ele permanece nesse processo até ser detectado a condição de repouso ou estabilidade, ou seja, em que a máquina encerrou a lavagem. Essa condição será explicada no tópico do MPU-6050.
- Finish: Estado final da lavagem, após ser detectado o encerramento dela. A partir desse estado, inicia-se o procedimento de notificação via email da finalização da lavagem.

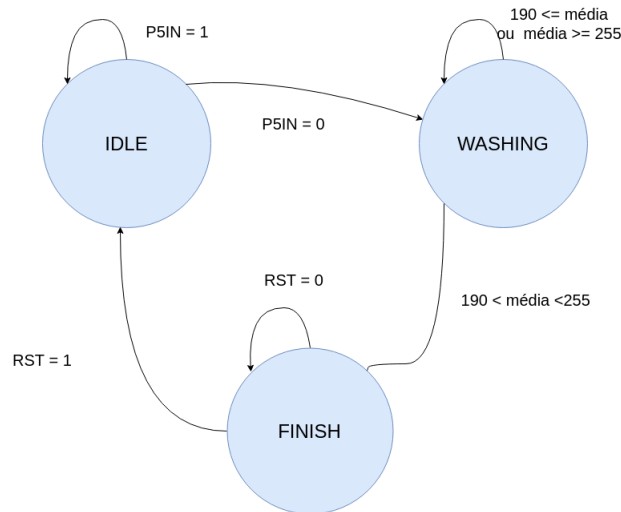


Figure 1: Diagrama de estados do dispositivo

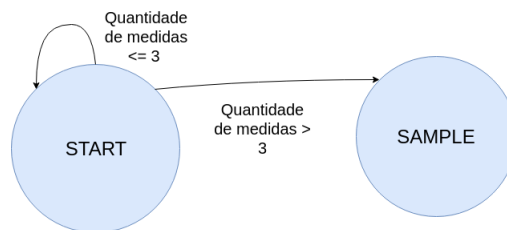
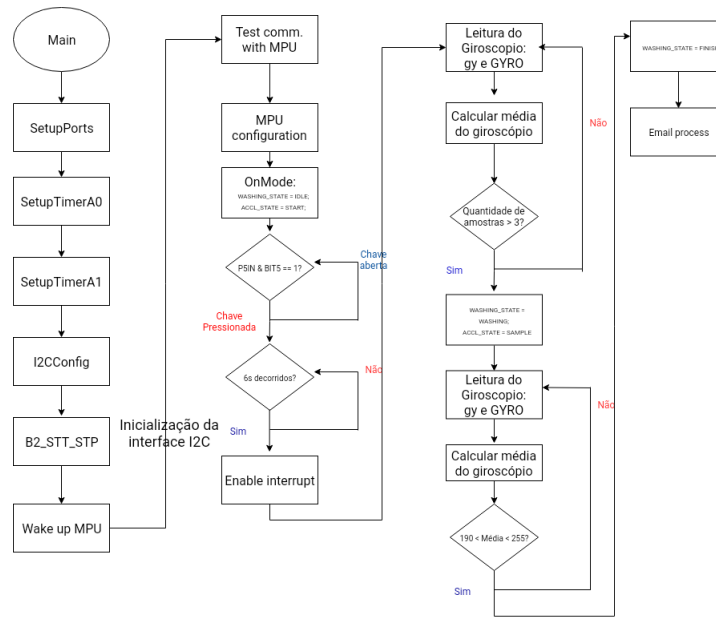


Figure 2: Diagrama de estados do acelerômetro

O algoritmo do programa principal, seguindo a descrição, é esquematizado pelo fluxograma abaixo:



## 2.1 MPU-6050 (Acelerômetro/Giroscópio)

Todo o procedimento de configuração que será descrito consta na documentação e datasheet do módulo MPU-6050. Recomenda-se a sua consulta para maior compreensão da descrição a seguir.

Inicialmente, o módulo MPU-6050 sempre é inicializado no modo *low power*. Para "acordá-lo", envia-se o byte 0x01 ao registrador do **power management**(0x6B).

Após sair do modo *low power*, testa-se o módulo comunicando-se com o registrador **Who Am I**, que sempre responde o endereço escravo do módulo (0x68).

Após isso, o módulo é resetado, retirado do modo *low power*; e é configurado via registrador de configuração (0x1A), colocando a taxa de coleta de valores do acelerômetro em 44 hz e giroscópio em 42 hz. A taxa de amostragem é definida via registrador 0x19, como sendo 200 hz seguindo a fórmula do datasheet. Então, é definido as escalas do acelerômetro e giroscópio, como sendo respectivamente, +2g e -2g; e +1000 graus/s e -1000 graus/s (via registradores 0x1C e 1B, respectivamente).

As leituras são realizadas via os registradores **YOUT high**(0x45) e **low**(0x46), par de bytes que compõe a velocidade angular medida no eixo y em escala de -32768 a 32767. Seu valor é convertido via regra de 3 para graus/s, e então disponibilizado ao programa.

Experimentalmente foi constatado que o intervalo lido pelo giroscópio no eixo y do sensor, na escala mencionada na configuração e **em repouso**; é entre 190 e 255 (levando em consideração o erro de medição do módulo). Para a finalidade do projeto, foi utilizado esse intervalo para rotular o estado da máquina de lavar roupa pelo seu tremor: a lavagem não está acontecendo se a leitura estiver nesse intervalo, que indica repouso da máquina. Caso contrário, o dispositivo interpretará como movimento, logo, a máquina está ainda no processo de lavagem(segundo a descrição do projeto). Por isso, esse intervalo de valores da velocidade angular do eixo y determina a ocorrência ou não da lavagem, levada em consideração na lógica do programa.

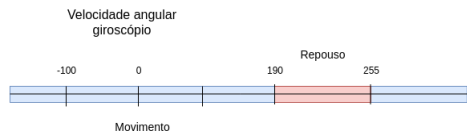


Figure 3: Intervalo de repouso do giroscópio

## 2.2 ESP-8266 01 (Módulo Wifi)

O ESP8266 foi desenvolvido pela Espressif Systems de Shanghai, como um Serial para SoC Wi-Fi (System On a Chip) baseado em um DPU Tensilica Xtensa LX3, ou seja, um adaptador Serial para Wi-Fi. O módulo pode ser

acionado através da interface serial UART usando o conjunto padrão de comandos AT. Cada conjunto de comandos contém quatro tipos de comandos AT.

Tipo	Formato	Descrição
<i>Test</i>	AT+ < X >=?	Consultar o comando Set ou parâmetros internos e seus valores de intervalo.
<i>Query</i>	AT+< X >?	Retorna o valor atual do parâmetro.
<i>Set</i>	AT+< X >=<... >	Defina o valor dos parâmetros definidos pelo usuário em comandos e executa.
<i>Execute</i>	AT+< X >	Executa comandos sem parâmetros definidos pelo usuário.

Os comandos AT devem estar em caixa alta e terminarem em  $\backslash r \backslash n$ . Todo o procedimento de configuração do ESP8266 e do MSP430 que será descrito consta na documentação e datasheet da placa e do módulo, destacando-se a Documentação AT Instruction Set. Recomenda-se a sua consulta para maior compreensão da descrição a seguir.

Foi usada a placa ESP-01 que possui os pinos usados para comunicação serial, RX e TX, bem como 4 pinos de controle, GPIO0 , GPIO2, CH\_PD e RST (reset), juntamente com VCC e GND.

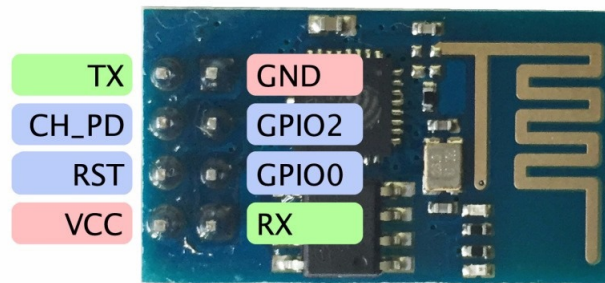


Figure 4: ESP8266 01

Para iniciar a comunicação e estabelecer algumas utilidades básicas, foram transmitidos os seguintes comandos:

- *AT* : Para começar e testar a comunicação.
- *AT+CIPCLOSE* : Fecha qualquer conexão anterior, TCP, UDP OU SSL.
- *AT+RST* : Reinicia o módulo.
- *AT+CWJAP="SSID","password"* : Acessa a internet
- *AT+CIFSR* : Obtém o endereço IP local.

Um computador cliente se comunica com um servidor SMTP (servidor de email) usando comandos SMTP. Portanto para enviar um email, além dos comandos AT, a mensagem enviada seguiu o modelo SMTP.

- *AT+CIPMUX=1* : Ativa várias conexões.
- *AT+CIPSERVER = 1, SRV\_PORT* : Cria o servidor TCP.
- *AT+CIPSTART=< linkID >,< type >,< remoteIP >,< remoteport >* : Estabelece Conexão TCP.
- *AT+CIPSEND = 4,NBYTES\_MESSAGE* : Envia dados do comprimento designado.

Em seguida a mensagem:

```
EHLO IP
AUTH LOGIN
EMAILBASE64 PASSWORD_BASE64
MAIL FROM:< "EMAILFROM" >
RCPT To:< "EMAILTO" >
DATA MESSAGE
QUIT
```

Todos os comandos foram enviados byte por byte e as respostas foram colocadas em um vetor através de uma rotina de interrupção do sinal recebido - *RXBUF*. A comunicação serial UART teve a seguinte configuração:

- BaudRate: 115200 (padrão)
- Sem bit de paridade
- 1 bit de Stop
- Prioridade para os bits menos significativos
- Clock: SMCLK
- Pinos: P6.0 e P6.1

### 3 Recursos utilizados

A placa utilizada para esse projeto foi o MSP430 FR5994. Dos seus recursos utilizado, consta:

1. GPIO(Pinos 1.1 e 1.2, correspondentes aos LEDs; pino 5.5 para a chave S2);
2. Timer A0;
3. Timer A1 e sua interrupção;
4. Interface de comunicação I2C, via pinos 7.1 e 7.0, SCL e SDA, respectivamente;
5. Módulo MPU-6050, via interface de comunicação I2C, alimentado em VCC por 3.3 V e GND pelo GND da placa;
6. Módulo Wifi ESP 8266 01 com comunicação serial UART, alimentado em VCC por 3.3 V e GND pelo GND da placa;