# Alimentação e preservação da bateria

Como já foi referido anteriormente, o sistema embebido irá se encontrar num contentor. Devido ao seu meio de localização, é quase impossível existir uma ligação de modo a este ter energia ilimitada. Por causa destas limitações, é necessário que o sistema embebido seja alimentado por uma bateria.

As baterias são uma fonte de energia muito limitada e pode trazer custos ao trocar estas (devido à energia ter acabado). Assim, foi posto em consideração na estrutura e implementação do sistema embebido técnicas de modo a preservar este bem limitado (fonte de alimentação).

No capítulo a seguir vamos falar sobre o sensor ultrassónico, sobre técnicas usadas para diminuir o seu gasto de energia, e depois de aplicadas essas técnicas, quanto é que iria consumir uma operação de uso deste periférico.

# Sensor Ultrassónico

O sensor ultrassónico [modelo] tem um consumo de energia considerável, sendo necessário 15 mA de corrente e 5V de tensão. Se este estive-se sempre operacional, iria haver um grande consumo de bateria, mesmo que o microcontrolador estivesse em modo *sleep*. Assim, para diminuir este gasto, foi colocado um MOSFET ao VCC do sensor, para controlar quando este recebe energia. É o microcontrolador que, com um porto do GPIO abre ou fecha a ligação do MOSFET.

A fonte de alimentação do sensor também irá ser a bateria, e foi feito um estudo para obter o consumo do sensor no seu tempo operacional.

Primeiramente, é necessário fornecer alimentação ao sensor, para isso, altera-se o MOSFET para deixar passar corrente. Depois disso é feito uma espera de 3 microssegundos para o sensor estabilizar. Depois deste estar estabilizado, é posto o Trigger com o valor lógico 1 e realiza-se uma espera de 10 microssegundos. No fim deste tempo, é colocado o Trigger com valor lógico ‘0’ e espera-se que o porto Echo fique com o valor lógico ‘0’. Mas este está dependente da chegada da onda sónica que foi enviada pelo o sensor, assim para este calculo foi usado o pior tempo possível. O pior tempo acontece quando a onda não volta para o sensor ou o tamanho do contentor é igual ao máximo que a onda consegue percorrer. O sensor no máximo consegue medir distâncias até 400 cm. Sabendo que a velocidade do som é 340 ms-1 o que equivale a uma espera máxima de 23200 microssegundos. No fim desliga-se o MOSFET para que corte a corrente ao sensor.

Feito um somatório sobre o tempo operacional do sensor, observa-se que este está operacional 23213 microssegundos (no pior dos casos) em cada uso, com um consumo constante de 15 mA.

# Sensor de Temperatura

O sensor de temperatura [modelo] já tem um modo de funcionamento que permite poupar o consumo de energia. Este modo chama-se parasite power, e a fonte de alimentação provêm do porto que é realizado a comunicação entre o sensor (porto DQ) e o microcontrolador (GPIO). Os portos VDD e o GND, do sensor, estão ligados ao ground. Quando o sensor encontrasse em modo Idle, este gasta 750 nA, despende 5 uA enquanto realiza comunicações com o master e 1,5mA em duas operações, salvar os dados na eeprom do sensor e guardar nos registos do sensor a temperatura do ambiente.

Para calcular a energia despendida do sensor de temperatura, não foi considerado o gasto da energia para a iniciação do sensor. Isto deve-se ao facto que esta ação só é realizada quando o microcontrolador inicia, o que equivale a uma vez em um/dois anos.

Como foi referido anteriormente, a implementação usada para saber se a temperatura ambiental encontra-se acima do normal, foi com a observação de uma flag do sensor, isto é, se o sensor quando acabar de converter a temperatura, se está não se situar dentro dos intervalos do alarme (os valores foram definidos na iniciação do sensor), este coloca uma flag com mo valor lógico a ‘1’. Posteriormente, com o envio de um comando Search Alarm, é possível saber se a temperatura encontra-se fora daqueles limites. Esta tecnica é mais rápida nas comunicações e menos despendedora de energia em comparação com a leitura da energia, o que seria necessário ler a ram do sensor.

A operação para observar se existe alarme, é composta pelos comandos Convert T e Search Alarm. Pode-se observar na Tabela X as operações para converter a temperatura e a sua duração.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reset | Skip room | Convert T | total | Unities |
| 1040 | 70\*8 | 70\*8 |  | microseconds |

De seguida o sensor convert a temperatura. Esta operação demora 93,75 ms (porque o sensor foi configurado a 9 bits) e existe um consumo de 1,5 mA durante esse tempo.

Por fim, é realizado um comando para observar se há alarme, este é composta por um conjunto de ações que se podem observar na Tabela Y.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reset | Search Alarm | Read Two Bits | total | Unities |
| 1040 | 70\*8 | 70\*2 |  | microseconds |

Com isto é possível observar que gasta X.

# Modulo Sigfox

O modulo Sigfox tem um consumo de 5 nA quando se encontra no estado OFF, no estado Idle 50 nA, e 33 mA quando está a transmitir uma.

# Microcontrolador

O microcontrolador LPC1769, é o que tem um maior consumo de energia. Durante o funcionamento tem um consumo de 50 mA [referencia]. De modo a reduzir este gasto, foram usadas técnicas que permitem colocar o processador em modo sleep. Neste modo, o processador não executa instruções, o que permite uma poupança no seu consumo. Enquanto o microcontrolador se encontrar no sleep mode, este gasta 2 mA.

Arredonda o tempo operacional dos sensores para 1 segundo, e supondo que um cidadão abre a tampa, e deixa-a aberta durante 9 segundos, e mais dois segundos para ler a medidas e comunicar com o controlador de Sigfox.

Em média, o dispositivo vai ter um gasto de 89,6 mA (50 do microcontrolador, 38 do controlador de Sigfox, 15 mA) durante 10 segundos.

E no estado Idle um gasto aproximado de 4 mA.

50 + 15 + 1,5 = durante um segundo

50 = durante 9 segundos

2m + 50 u = Idle

5 + 33 = durante 10 segundos

3 m + 90 u = durante Idle